

Apple

\$1.50



Assembly Line

Volume 4 -- Issue 1

October, 1983

In This Issue...

Compilation of Monitor Modifications	2
Still More Tinkering with VCR.	11
Corrections to the Generic Screen Dump	12
Index to Volumes 1-3	Insert
Price Changes.	13
Duplicated Ideas and Red Faces	13
Faster Booting for ScreenWriter II	14
Large Assembly Listing into Text File.	16
Avoiding EXTRA DEFINITION ERROR.	17
Lower Case Titles in Version 1.1	17
Suppressing Unwanted Object Bytes.	19
Where To?.	19
Macro-Calculated Spiral Screen Clear	20
Counting Lines	22

Index to Apple Assembly Line

Why haven't we ever published an index to AAL?, you ask. Now that there are three year's worth of back issues to dig through for that article you know you saw a while back, wouldn't a true index come in handy? Well here it is! The 12 center pages of this issue are a complete index to volumes 1 through 3 of Apple Assembly Line. That's October, 1980 through September, 1983, all at your fingertips. The index is placed in the center of this issue so that, if you wish, you can easily remove those pages and store them separately.

More Applesoft Variable Cross Reference

In this issue Louis Pitz presents us with still more tinkering with the old Applesoft Variable Cross Reference. Now that the program has been modified a couple of times, and since it appeared way back in the second issue of Apple Assembly Line, we'll include the complete source code, including all of Louis' enhancements, on the next Quarterly Disk. Remember that all of the back issues are still available, if you don't have Volume 1, Number 2.

A Compilation of Monitor Modifications.....Steve Knouse

Over the years since I bought my Apple I have been collecting various handy modifications to the Apple Monitor. I wanted a convenient way to load up all my patches so that they would be there when I needed them.

Let me point out right now that the following set of patches will NOT work in an Apple //e. They are only for the Apple II Plus monitor. Anyway, several of my favorite patches are already implemented in the Apple //e; the others may fit, but I haven't tried them.

There are two basic ways to get a modified monitor into an Apple. The first requires burning an EPROM with the new version, modifying the motherboard to accept an EPROM in the F8 ROM socket, and plugging it in. (Rather than cutting and splicing the motherboard, a better way is to use a PROMETTE from Computer Micro Works.) The second way is to run out of a language card (16K RAM Card), with a modified monitor at F800 in the RAM card. Some RAM cards may not allow this, leaving the motherboard F8 ROM always switched on, but all the ones I have tried work. If you want to use Applesoft with the modified monitor, or patch Applesoft as well, you can copy it up into the language card too.

I combined my favorite patches with Bill Morgan's patch program (see "PATCHER: General Purpose Patch Installer", AAL, April, 1983) so that BRUNning the program copies the motherboard monitor into a RAM card and then installs all the patches.

The listing that follows uses the .PH and .EP directives found in Version 1.1 of the S-C Macro Assembler. .PH starts a phase, and .EP ends one. At the start of a phase the current assembler origin is saved and the address from the .PH is substituted. Code continues to be assembled into the target file or at the target address, and the saved origin is incremented along with the phase origin. At the end of the phase the saved origin is restored. This allows me to assemble a series of patches with the correct addresses all into one big target file.

Here is a list of my favorite patches:

- 1 Allow lowercase input -- nullify conversion of lowercase to uppercase, make cursor over lowercase character to uppercase inverse (since Apple doesn't have inverse or flashing lowercase). (From Videx Keyboard Enhancer II Manual, page 4.)
- 2 Non-flashing cursor -- Make cursor inverse instead of flashing. (From Videx Keyboard Enhancer II Manual, page 4.)
- 3 Inverse + cursor when in escape mode -- to indicate IJKM is active. (By Donald W. Miller, Jr., Call-APPLE Mar 83 pp 51-52.)

- 4 ASCII dump -- display both hex and ASCII values. (By Peter Bartlett, AAL Dec 81 pp 18-20, and Bruce Field, AAL Jul 83 page 20.)
- 5 Mask -- XYYY<ADR1.ADR2W masks bytes in memory range, ANDing with XX and ORing with YY. (By Bob Sander-Cederlof, AAL Dec 82 pp 10-11.)
- 6 Search -- XYYY<ADR1.ADR2S searches memory range for XYYY, printing addresses of matches. If XYYY is in the range \$00-\$FF, only one byte will be compared; otherwise both bytes will be compared during the search. (By Steve Knouse)

I included several conditional assembly options, using the .DO, .ELSE, and .FIN directives. These let you select or reject the non-flashing cursor patch and the lowercase display patch. The third option allows you to copy Applesoft from the motherboard along with the monitor, or just the monitor by itself.

```

1000 *SAVE S.KNOUSE'S MONITOR PATCHES
1010 *-----
1020 *
1030 *   A COMPILATION OF MONITOR MODIFICATIONS
1040 *-----
0001- 1050 YES .EQ 1
0000- 1060 NO .EQ 0
      1070 *
      1080 *   OPTIONS
      1090 *
0001- 1100 NFC .EQ YES SET TO YES IF YOU WANT
      1110 * A NON-FLASHING CURSOR
0001- 1120 LOWERCASE .EQ YES SET TO YES IF YOU CAN
      1130 * DISPLAY LOWER CASE
0001- 1140 W.APPLESOFT .EQ YES SET TO YES IF YOU WANT
      1150 * TO MOVE APPLESOFT WITH
      1160 * THE MONITOR, ELSE SET
      1170 * TO NO IF YOU ONLY WANT
      1180 * TO MOVE AND MODIFY THE
      1190 * MONITOR
      1200 *-----
0000- 1210 PNTR .EQ $00,01
0002- 1220 PATCH .EQ $02,03
003C- 1230 A1L .EQ $3C
003D- 1240 A1H .EQ A1L+1
003E- 1250 A2L .EQ $3E
003F- 1260 A2H .EQ A2L+1
0042- 1270 A4L .EQ $42
0043- 1280 A4H .EQ A4L+1
0028- 1290 BASL .EQ $28
0024- 1300 CH .EQ $24
0038- 1310 KSWL .EQ $38
      1320 *-----
FDED- 1330 COUT .EQ $FDED
FEF6- 1340 CRMON .EQ $FEF6
FD8E- 1350 CROUT .EQ $FD8E
FCC9- 1360 MON.HEADR .EQ $FCC9
FE2C- 1370 MON.MOVE .EQ $FE2C
FCBA- 1380 NXTA1 .EQ $FCBA
FD92- 1390 PRA1 .EQ $FD92
FDDA- 1400 PRBYTE .EQ $FDDA
FF2D- 1410 PRERR .EQ $FF2D
FDOC- 1420 RDKEY .EQ $FDOC
FEFD- 1430 MON.READ .EQ $FEFD
FECD- 1440 MON.WRITE .EQ $FECD
      1450 *-----
C081- 1460 ROMR.RAMW .EQ $C081
C083- 1470 RAMRW .EQ $C083
      1480 *-----
      1490 *

```

```

1500 * GENERAL PURPOSE PATCHER
1510 *
1520 *-----
1530 .OR $2D0
1540 .DO W.APPLESOFT
1550 .TF PATCH MONITOR & APPLESOFT
D000- 1560 MON.START .EQ $D000
1570 .ELSE
1580 .TF PATCH MONITOR ONLY
1590 MON.START .EQ $F800
1600 .FIN
FFFF- 1610 MON.END .EQ $FFFF
1620 *-----
1630 PATCH.MONITOR
02D0- A9 00 1640 LDA #MON.START COPY MONITOR TO RAM CARD
02D2- 85 42 1650 STA A4L
02D4- 85 3C 1660 STA A1L
02D6- A9 D0 1670 LDA /MON.START
02D8- 85 43 1680 STA A4H
02DA- 85 3D 1690 STA A1H
02DC- A9 FF 1700 LDA #MON.END
02DE- 85 3E 1710 STA A2L
02E0- A9 FF 1720 LDA /MON.END
02E2- 85 3F 1730 STA A2H
02E4- AD 81 CO 1740 LDA ROMR.RAMW WRITE ENABLE RAM CARD
02E7- AD 81 CO 1750 LDA ROMR.RAMW BY 2 OF THESE
02EA- A0 00 1760 LDY #0 SET UP MON.MOVE
02EC- 20 2C FE 1770 JSR MON.MOVE COPY FROM MOTHERBOARD TO RAMCARD
1780 *
02EF- A9 25 1790 LDA #PATCHES-1
02F1- 85 00 1800 STA PNTR
02F3- A9 03 1810 LDA /PATCHES-1
02F5- 85 01 1820 STA PNTR+1
02F7- A0 C0 1830 LDY #0
1840 *
02F9- 20 1D 03 1850 .1 JSR GET.BYTE LENGTH OF NEXT PATCH
02FC- F0 1B 1860 BEQ .4
02FE- AA 1870 TAX SAVE LENGTH IN X
02FF- 20 1D 03 1880 JSR GET.BYTE GET ADDR OF PATCH
0302- 85 02 1890 STA PATCH
0304- 20 1D 03 1900 JSR GET.BYTE
0307- 85 03 1910 STA PATCH+1
1920 *
0309- 20 1D 03 1930 .2 JSR GET.BYTE GET A BYTE
030C- 91 02 1940 STA (PATCH),Y STORE AT DESTINATION
030E- E6 02 1950 INC PATCH BUMP SOURCE ADDRESS
0310- D0 02 1960 BNE .3
0312- E6 03 1970 INC PATCH+1
0314- CA 1980 .3 DEX DECREMENT NUMBER OF BYTES
0315- D0 F2 1990 BNE .2 LOOP FOR MORE
0317- F0 E0 2000 BEQ .1 ... ALWAYS
2010 *
0319- AD 83 CO 2020 .DO W.APPLESOFT
031C- 60 2030 LDA RAMRW
2040 .4 RTS
2050 .ELSE
2060 .4 RTS
2070 .FIN
2080 *-----
031D- E6 00 2090 GET.BYTE
031F- D0 02 2100 INC PNTR
0321- E6 01 2110 BNE .1
0323- B1 00 2120 INC PNTR+1
0325- 60 2130 .1 LDA (PNTR),Y
2140 RTS
2150 *-----
2160 .MA PATCH
2170 ]1.ORG .EQ ]2
2180 .DA #]1.LENGTH
2190 .DA ]1.ORG
2200 .PH ]1.ORG
2210 ]1
2220 .EM
2230 *
2240 .MA ENDP
2250 ]1.END .EQ #-1
2260 ]1.LENGTH .EQ #-]1
2270 .EP
2280 .EM
2290 *-----

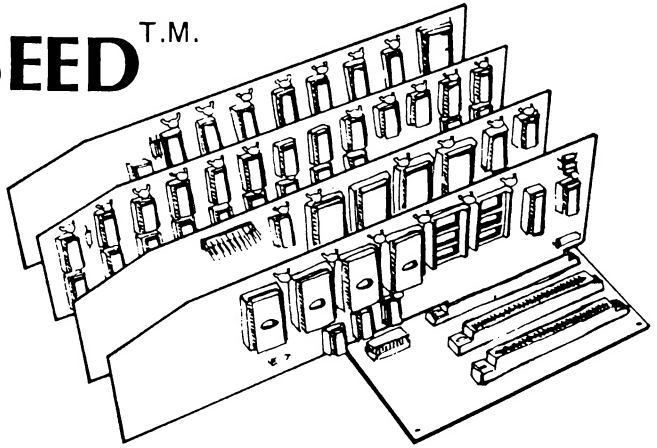
```

```

0326-      2300 PATCHES      .EQ *
          2310 *-----*
          2320 * MONITOR LOWERCASE INPUT ROUTINE
          2330 *-----*
          2340 *
          2350 *---DON'T STOMP ON LOWERCASE-----
0326-      2360 >PATCH NOP.CONVERT,$FD82
FD82-      0000> NOP.CONVERT.ORG .EQ $FD82
0326- 02      0000> .DA #NOP.CONVERT.LENGTH
0327- 82 FD      0000> .DA NOP.CONVERT.ORG
          0000> .PH NOP.CONVERT.ORG
          0000> NOP.CONVERT
          2370 AND #$FF DO NOTHING
FD84-      2380 >ENDP NOP.CONVERT
FD83-      0000> NOP.CONVERT.END .EQ *-1
0002-      0000> NOP.CONVERT.LENGTH .EQ *-NOP.CONVERT
          0000> .EP
          2390 *
          2400 *---MAKE SENSIBLE CURSOR-----*
          2410 >PATCH HANDLE.CURSOR,$FBB3
032B-      0000> HANDLE.CURSOR.ORG .EQ $FBB3
FBB3-      0000> .DA #HANDLE.CURSOR.LENGTH
032B- 0C      0000> .DA HANDLE.CURSOR.ORG
032C- B3 FB      0000> .PH HANDLE.CURSOR.ORG
          0000> HANDLE.CURSOR
          2420 CMP #$E0 IS IT LOWER CASE?
          2430 BCS .1
          2440 AND #$3F NO - MAKE CHAR INVERSE
          2450 DO NFC
          2460 ORA #$00
          2470 ELSE
          2480 ORA #$40 THEN FLASHING
          2490 .FIN
          2500 RTS
          2510 .1 AND #$1F CONVERT TO UC INVERSE
          2520 RTS
          2530 >ENDP HANDLE.CURSOR
          0000> HANDLE.CURSOR.END .EQ *-1
          0000> HANDLE.CURSOR.LENGTH .EQ *-HANDLE.CURSOR
          0000> .EP
          2540 *
          2550 *---CALL NEW CURSOR ROUTINE-----*
          2560 >PATCH VEC.HANDLE.CURSOR,$FD11
033A-      0000> VEC.HANDLE.CURSOR.ORG .EQ $FD11
FD11-      0000> .DA #VEC.HANDLE.CURSOR.LENGTH
033A- 04      0000> .DA VEC.HANDLE.CURSOR.ORG
033B- 11 FD      0000> .PH VEC.HANDLE.CURSOR.ORG
          0000> VEC.HANDLE.CURSOR
          2570 JSR HANDLE.CURSOR GO TO PATCH
          2580 NOP FILL BYTE
          2590 >ENDP VEC.HANDLE.CURSOR
          0000> VEC.HANDLE.CURSOR.END .EQ *-1
          0000> VEC.HANDLE.CURSOR.LENGTH .EQ *-VEC.HANDLE.CURSOR
          0000> .EP
          2600 *-----*
          2610 * ASCII DUMP
          2620 *-----*
          2630 *
          2640 *---MODIFIED DUMPER-----*
          2650 >PATCH ASC.DUMP,MON.HEADR
0341-      0000> ASC.DUMP.ORG .EQ MON.HEADR
FCC9-      0000> .DA #ASC.DUMP.LENGTH
0341- 21      0000> .DA ASC.DUMP.ORG
0342- C9 FC      0000> .PH ASC.DUMP.ORG
          0000> ASC.DUMP
          2660 PHA SAVE CHAR
          2670 LDA A1L GET LO ADDR BYTE
          2680 AND #$07 MOD 8
          2690 CLC ADD DISPLACEMENT
          2700 ADC #30 OF 30 CHAR
          2710 TAY
          2720 PLA RECOVER CHARACTER
          2730 PHA SAVE IT AGAIN
          2740 ORA #$80 FORCE NORMAL VIDEO
          2750 CMP #$A0 MAKE CONTROL CHAR INVERSE
          2760 BCS .1 ...NOT CONTROL
          2770 DO LOWERCASE
          2780 AND #$7F ...CONTROL
          2790 .1 STA (BASL),Y PUT ON SCREEN
          2800 NOP TO STAY ALIGNED W/
          2810 NOP NON-LOWERCASE CODE

```

APPLESEED^{T.M.}



Appleseed is a complete computer system. It is designed using the bus conventions established by Apple Computer for the Apple II. Appleseed is designed as an alternative to using a full Apple II computer system. The Appleseed product line includes more than a dozen items including CPU, RAM, EPROM, UART, UNIVERSAL Boards as well as a number of other compatible items. This ad will highlight the Mother board.

BX-DE-12 MOTHER BOARD

The BX-DE-12 Mother board is designed to be fully compatible with all of the Apple conventions. Ten card slots are provided. Seven of the slots are numbered in conformance with Apple standards. The additional three slots, lettered A, B and C, are used for boards which don't require a specific slot number. The CPU, RAM and EPROM boards are often placed in the slots A, B and C.

The main emphasis of the Appleseed system is illustrated by the Mother Board. The absolute minimum amount of circuitry is placed on the Mother Board; only the four ICs which are required for card slot selection are on the mother board. This approach helps in packaging (flexibility & smaller size), cost (buy only what you need) and repairability (isolate and fix problems through board substitution).

Appleseed products are made for O.E.M.s and serious industrial/scientific users. Send for literature on the full line of Appleseed products; and, watch here, each month, for additional items in the Appleseed line.

Appleseed products are not sold through computer stores.

Order direct from our plant in California.

Apple is a registered trademark of Apple Computer, Inc.

DOUGLAS ELECTRONICS

718 Marina Blvd., San Leandro, CA 94577 • (415) 483-8770

```

FCE0- EA      2820      NOP
FCE1- EA      2830      NOP
FCE2- EA      2840      NOP
FCE3- EA      2850      NOP
                2860      .ELSE
                2870      LDA # $DF      MAKE CTRL-CHARS INVERSE
                2880      CMP # $E0      IN LOWER CASE RANGE?
                2890      BCC .2      ..NO, DISPLAY NORMAL VIDEO
                2900      AND # $1F      ..YES, FORCE INVERSE VIDEO
                2910      .2      STA (BASL),Y STORE IT ON SCREEN
                2920      .FIN
FCE4- AO 00    2930      LDY #0      RESTORE Y REG
FCE6- 68      2940      PLA      RECOVER BYTE AGAIN
FCE7- 4C DA FD 2950      JMP PRBYTE
FCEA-         2960      >ENDP ASC.DUMP
FCE9-         0000> ASC.DUMP.END .EQ #-1
0021-         0000> ASC.DUMP.LENGTH .EQ #-ASC.DUMP
                0000> .EP
                2970      *
                2980      *---CALL ASCII DUMP-----
0365-         2990      >PATCH VEC.ASC.DUMP,$FDBD
FDBD-         0000> VEC.ASC.DUMP.ORG .EQ $FDBD
0365- 03      0000> .DA #VEC.ASC.DUMP.LENGTH
0366- BD FD    0000> .DA VEC.ASC.DUMP.ORG
                0000> .PH VEC.ASC.DUMP.ORG
                0000> VEC.ASC.DUMP
FDBD- 20 C9 FC 3000      JSR ASC.DUMP
FDC0-         3010      >ENDP VEC.ASC.DUMP
FDBF-         0000> VEC.ASC.DUMP.END .EQ #-1
0003-         0000> VEC.ASC.DUMP.LENGTH .EQ #-VEC.ASC.DUMP
                0000> .EP
                3020      *-----
                3030      * + CURSOR IN ESCAPE MODE
                3040      *-----
                3050      *
                3060      *---SAVE SCREEN, SPOT + -----
036B-         3070      >PATCH RDKEY2,ASC.DUMP.END+1
FCEA-         0000> RDKEY2.ORG .EQ ASC.DUMP.END+1
036B- 22      0000> .DA #RDKEY2.LENGTH
036C- EA FC    0000> .DA RDKEY2.ORG
                0000> .PH RDKEY2.ORG
                0000> RDKEY2
FCEA- A4 24    3080      LDY CH      SAVE CHARACTER
FCEC- B1 28    3090      LDA (BASL),Y
FCEE- 48      3100      PHA
FCEF- A9 2B    3110      LDA #' +      PUT AN INVERSE + ON SCREEN
FCF1- 91 28    3120      STA (BASL),Y
FCF3- 68      3130      PLA      GET THE CHARACTER BACK
FCF4- 6C 38 00 3140      JMP (KSWL)
FCF7-         3150      .BS RDKEY-# FILL W/ 0'S TO RDKEY
FDOC-         3160      >ENDP RDKEY2
FDOB-         0000> RDKEY2.END .EQ #-1
0022-         0000> RDKEY2.LENGTH .EQ #-RDKEY2
                0000> .EP
                3170      *
                3180      *---CALL + CURSOR-----
0390-         3190      >PATCH VEC.RDKEY2.1,$FBA2
FBA2-         0000> VEC.RDKEY2.1.ORG .EQ $FBA2
0390- 03      0000> .DA #VEC.RDKEY2.1.LENGTH
0391- A2 FB    0000> .DA VEC.RDKEY2.1.ORG
                0000> .PH VEC.RDKEY2.1.ORG
                0000> VEC.RDKEY2.1
FBA2- 20 EA FC 3200      JSR RDKEY2
FBA5-         3210      >ENDP VEC.RDKEY2.1
FBA4-         0000> VEC.RDKEY2.1.END .EQ #-1
0003-         0000> VEC.RDKEY2.1.LENGTH .EQ #-VEC.RDKEY2.1
                0000> .EP
                3220      *
                3230      *---CALL + CURSOR-----
0396-         3240      >PATCH VEC.RDKEY2.2,$FD2F
FD2F-         0000> VEC.RDKEY2.2.ORG .EQ $FD2F
0396- 03      0000> .DA #VEC.RDKEY2.2.LENGTH
0397- 2F FD    0000> .DA VEC.RDKEY2.2.ORG
                0000> .PH VEC.RDKEY2.2.ORG
                0000> VEC.RDKEY2.2
FD2F- 20 EA FC 3250      JSR RDKEY2
FD32-         3260      >ENDP VEC.RDKEY2.2
FD31-         0000> VEC.RDKEY2.2.END .EQ #-1
0003-         0000> VEC.RDKEY2.2.LENGTH .EQ #-VEC.RDKEY2.2
                0000> .EP

```

```

3270 *-----
3280 *   MASK BIT CONTROL OVER MEMORY RANGE
3290 *   XYYY<ADR1.ADR2W   FORMS M=(M.AND.XX).OR.YY
3300 *-----
3310 *
039C- 3320 >PATCH WRITE,MON.WRITE
FECD- 0000> WRITE.ORG .EQ MON.WRITE
039C- 29 0000> .DA #WRITE.LENGTH
039D- CD FE 0000> .DA WRITE.ORG
0000> .PH WRITE.ORG
0000> WRITE
FECD- B1 3C 3330 LDA (A1L),Y GET A BYTE
FECD- 25 43 3340 AND A4H AND IT WITH XX
FED1- 05 42 3350 ORA A4L OR IT WITH YY
FED3- 91 3C 3360 STA (A1L),Y PUT IT BACK
FED5- 20 BA FC 3370 JSR NXTA1 INCR ADDRESS
FED8- 90 F3 3380 BCC WRITE LOOP FOR MORE
FEDA- 60 3390 RTS
FEDE- 3400 .BS CRMON-* FILL W/ 0'S TO CRMON
FEF6- 3410 >ENDP WRITE
FEF5- 0000> WRITE.END .EQ *-1
0029- 0000> WRITE.LENGTH .EQ *-WRITE
0000> .EP
3420 *-----
3430 *   SEARCH
3440 *   XYYY<ADR1.ADR2S
3450 *-----
3460 *
03C8- 3470 *---SEARCH PROCESSOR-----
FEFD- 3480 >PATCH SEARCH,MON.READ
03C8- 30 0000> SEARCH.ORG .EQ MON.READ
03C9- FD FE 0000> .DA #SEARCH.LENGTH
0000> .DA SEARCH.ORG
0000> .PH SEARCH.ORG
0000> SEARCH
FEFD- A5 43 3490 LDA A4H IS THIS A 1 OR 2 BYTE COMPARE
FEFF- F0 08 3500 BEQ .2 ..ONE BYTE
FF01- A5 3E 3510 LDA A2L ..TWO BYTE
FF03- D0 02 3520 BNE .1 DECREMENT ENDING ADDR
FF05- C6 3F 3530 DEC A2H
FF07- C6 3E 3540 DEC A2L
3550 *
FF09- A5 43 3560 .2 LDA A4H GET FIRST BYTE TO COMPARE
FF0B- F0 05 3570 BEQ .3 IF ZERO DO A ONE BYTE SEARCH
FF0D- D1 3C 3580 CMP (A1L),Y COMPARE WITH MEMORY
FF0F- D0 0C 3590 BNE .4 NOT EQUAL - GO TO NEXT BYTE
FF11- C8 3600 INY GET NEXT BYTE
FF12- B1 3C 3610 .3 LDA (A1L),Y
FF14- A0 00 3620 LDY #0 RESTORE Y REG
FF16- C5 42 3630 CMP A4L COMPARE
FF18- D0 03 3640 BNE .4 NOT EQUAL - DRIVE ON
FF1A- 20 92 FD 3650 JSR PRA1
FF1D- 20 BA FC 3660 .4 JSR NXTA1 GET NEXT BYTE
FF20- 90 E7 3670 BCC .2 LOOP FOR MORE
FF22- 60 3680 RTS
FF23- 3690 .BS PRERR-* FILL W/ 0'S TO PRERR
FF2D- 3700 >ENDP SEARCH
FF2C- 0000> SEARCH.END .EQ *-1
0030- 0000> SEARCH.LENGTH .EQ *-SEARCH
0000> .EP
3710 *
03FB- 3720 *---PATCH COMMAND TABLE-----
FFDE- 3730 >PATCH VEC.SEARCH,$FFDE
03FB- 01 0000> VEC.SEARCH.ORG .EQ $FFDE
03FC- DE FF 0000> .DA #VEC.SEARCH.LENGTH
0000> .DA VEC.SEARCH.ORG
0000> .PH VEC.SEARCH.ORG
0000> VEC.SEARCH
FFDE- EC 3740 .DA #$EC 'S' EOR $B0 + $89
FFDF- 3750 >ENDP VEC.SEARCH
FFDE- 0000> VEC.SEARCH.END .EQ *-1
0001- 0000> VEC.SEARCH.LENGTH .EQ *-VEC.SEARCH
0000> .EP
3760 *-----
03FF- 00 3770 .DA #0 END OF PATCHES
3780 *-----
03FF- 3790 END .EQ *-1
0130- 3800 LENGTH .EQ END-PATCH.MONITOR+1

```

D O W N L O A D I N G C U S T O M C H A R A C T E R S E T S

One of the features 'hidden' in many printers available today is their ability to accept user-defined character sets. With the proper software, these **custom characters** are 'downloaded' from your Apple II computer to the printer in a fraction of a second. Once the printer has 'learned' these new characters, they will be remembered until the printer is turned off.

After the downloading operation, you can use your printer with virtually any word processor. Just think of the possibilities! There's nothing like having your own **CUSTOM CHARACTERS** to help convey the message. And you still have access to those built-in fonts as well! Here's a quick look at some possible variations:

BUILT-IN

10CPI: AaBbCcDdEeFfGgHhIiJjKk
12CPI: AaBbCcDdEeFfGgHhIiJjKk
16CPI: AaBbCcDdEeFfGgHhIiJjKk

5CPI: AaBbCcDdEeFf
6CPI: AaBbCcDdEeFf
8CPI: AaBbCcDdEeFf

CUSTOM

AaBbCcDdEeFfGgHhIiJjKk
AaBbCcDdEeFfGgHhIiJjKk
AaBbCcDdEeFfGgHhIiJjKk

AaBbCcDdEeFf
AaBbCcDdEeFf
AaBbCcDdEeFf

And let's not forget Enhanced and Underlined printing as well...

AaBbCcDdEeFfGgHhIiJjKk
AaBbCcDdEeFfGgHhIiJjKk

AaBbCcDdEeFfGgHhIiJjKk
AaBbCcDdEeFfGgHhIiJjKk

The Font Downloader & Character Editor software package has been developed by RAK-WARE to help you unleash the power of your printer. The basic package includes the downloading software with 4 fonts to get you going. Also included is a character editor so that you can turn your creativity loose. Use it to generate unique character fonts, patterns, symbols and graphics. A detailed user's guide is provided on the program diskette.

SYSTEM REQUIREMENTS:

- * APPLE II, APPLE II Plus, APPLE //e or lookalike with 48K RAM
- * 'DUMB' Parallel Printer Interface Board (like Apple's Parallel Printer Interface, TYMAC's PPC-100 or equivalent)

The Font Downloader & Editor package is only \$39.95 and is currently available for either the Apple Dot Matrix Printer or C.Itoh 8510AP (specify printer). Epson FX-80 and Okidata versions coming soon. Enclose payment with order to avoid \$3.00 handling & postage charge.

R A K - W A R E
41 Ralph Road West Orange New Jersey 07052

Say You Saw It In **APPLE ASSEMBLY LINE!**

I finally figured out how to modify the Applesoft Variable Cross Reference (from the November, 1980 AAL) to distinguish between defined functions and array variables. As Bob mentioned at that time, VCR tags an occurence of FN AB() as an appearance of the array variable AB().

It turns out that the changes needed aren't many, and are compatible with my tinkering in the August '83 AAL, which added 80-column output to a printer.

As VCR is scanning for variables, in the GET.NEXT.VARIABLE section, add the check for the FN token in lines 2132-2134. If found, go to lines 2222-2228 to set a flag and go back to get the NEXT.CHAR.NOT.QUOTE. Unless the Applesoft program is in error, a variable name immediately follows the FN token.

In PACK.VARIABLE.NAME, the program distinguishes variables by VARNAM+2 having a space, \$, or %. Array variables have the high bit set. In lines 2791-2796 I set apart FN variables by placing a dash (-) with the high bit set in VARNAM+2. This will make FN types come after the others alphabetically.

Now we come to the printing stage, in PRINT.LETTER.CHAIN. There the variable name (and dash, in case of FN types) is printed. If the high bit of VARNAM+2 is set, lines 4292-4294 check for the dash value. If so, skip to lines 4511-4515 and print out "FN" also.

This way, FN AB will come out as "AB-FN", which is a bit of a cop-out on my part. But I opted for making minimal changes to VCR to keep things simple.

```
2132      CMP #$C2
2134      BEQ .4

2222 .4   STA $7          set FLAG2
2224      BEQ .1          ...always
2226 *    unless syntax error, NEXT.CHAR.NOT.QUOTE
2228 *    will be letter, hence variable!

2791      LDA $7          recall FLAG2
2792      CMP #$C2         FN token?
2793      BNE .5           (to RTS)
2794      LDA #'--80       "-"
2795      STA VARNAM+2     to indicate FN
2796      STA $7          and reset FLAG2

4292      CMP #$AD        not array, but FN?
4294      BEQ .6

4511 .6   LDA #'F        add 'FN' after
4512      JSR PRINT.CHAR
4513      LDA #'N          variable name
4514      JSR PRINT.CHAR
4515      BNE .4          ...always
```

Corrections to the Generic Screen Dump

Steve Knouse called to thank us for printing his Generic Screen Dump program last month, and to chew us out for garbling it.

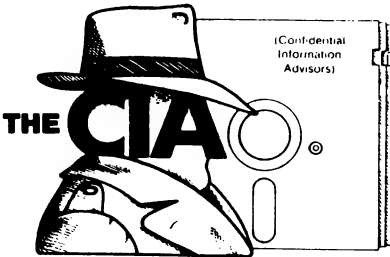
It seems that we edited and renumbered the code, but didn't update the line number references in the text.

Here's a table to translate what the article says into what it means:

Says	Means
1610	1100
2030	1460
2190	1620
2250	1680
2260	1690
2270	1700
2280	1710
2290	1720
2300	1730
2310	1740

Sorry about that, readers. Sorry about that, Steve.

[And another last-minute correction -- the TAY instruction in line 1510 should be a TYA.]



(Confidential Information Advisors)

Why buy Commercial Software and not know what's going on?

Meet 'TRICKY DICK' and his accomplices:-
The LINGUIST
The CODE BREAKER
The TRACER
The TRACKER

Why not buy a kit of 5 powerful disk espionage utilities, all on one disk, that allow you to investigate, edit, locate list, trace, rescue, translate, patch, repair, verify, examine, protect, unprotect, encrypt, decrypt, and analyse programs or textfiles on normal and PROTECTED disks. This kit includes the 'CIA FILES', a 60,000+ word book on all aspects of the Apple (R) disk, many of which have never before appeared in print. "The CIA Files" gives you extensive, detailed, easy to read, information on disk repair, file recovery, DOS patches, copy protection, disk formatting, program encryption, and many other vital topics, showing you keystroke by keystroke, EXACTLY how to do it!

CHECKSUMS:
defeat them, Y/N?

HEADERS:
change them as you like -- even insert 'wild cards'!

WRITE BACK TO ANY DISK!
(You need an Apple II+™ or IIe™ 48K and 1 drive)

TRAILERS: modify at will, wild cards if you please!

LIST: in APPLESOFT, INTEGER, ASSEMBLY LANGUAGE direct from the nibble dump of a protected disk!

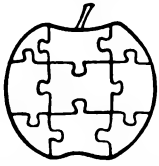
Specify ENCODING 6 & 2, 5 & 3, 4 & 4,

TRICKY DICK THE LINGUIST BY T TSE

D596AA Y AADE DOS SL=6 T=23 <00> VOL
D6ABAE N 0000 3,3 DR=1 S=0F <--> 254
D5A9AD 0 DEAAEB PR=0 <AL> <62> <H>

Send \$65.00 Money Order
(Checks, allow time to clear)
to:- GOLDEN DELICIOUS
SOFTWARE LTD.,
(Sales Dept. A3),
350 Fifth Ave., Suite 3308
New York, N.Y. 10001

Apple



Assembly

Line

INDEX

Volumes 1-3

October, 1980

through

September, 1983

Index to Articles in "Apple Assembly Line", Volumes 1-3

AAAA

Amper-Monitor.....	Bob S-C...	9/83/14-16
Amperсанд Monitor Caller.....	Bob S-C...	6/83/30-32
Apple Chips.....	Bob S-C & Bill Morgan...	5/83/12
Applesoft		
Adding Decimal ASCII Strings.....	Bob S-C...	2/83/2-11
All About PTRGET and GTARYPT.....	Bob S-C...	3/83/2-9
A New Hi-Res Function (HXPLT).....	Mike Laumer...	6/82/7-10
Applesoft Program Locator.....	Bill Morgan...	11/82/19-22
Applesoft Source, Completely Documented.....	Bob S-C...	12/82/15
CHRGOT and CHRGET in Applesoft.....	Bob S-C...	9/81/8-9
Correction to "CHRGOT...".....	Bob S-C...	10/81/18
Compute GOSUB.....	Bob S-C...	1/81/8
EXEC without END from Applesoft.....	Bob S-C...	11/82/17
Fast String Input Routine.....	Bob S-C...	4/81/6-8
Correction to "Fast String Input".....	Bob S-C...	10/81/18
Improved "Fast String Input".....	Bob S-C...	12/81/16-17
Field Input Routine.....	Bob Potts...	9/81/2-7
Finding Applesoft Line Numbers.....	Bob Potts...	8/81/2
Floating Point Number Format.....	Bob S-C...	11/81/2
Formatted Print Routine.....	Bob S-C...	11/81/6-13
Garbage Collection Indicator for Applesoft.....	Lee Meador...	3/83/22
Generate Machine Code with Applesoft.....	Bob S-C...	9/83/10-12
GOTO from Assembly Language.....	Bob S-C...	12/81/23-24
Hex Constants in Applesoft.....	David Bartley...	12/81/6-9
Hi-Res SCRN Function.....	Bob S-C...	5/81/2-3
Hi-Res SCRN Function with Color.....	David Doudna...	1/82/2-5
Hi-Res Subroutines.....	Bob S-C...	12/81/2-4
Integer Input (0-65535) Using ROM Routines.....	Peter Meyer...	12/81/insert
Internal Entry Points.....	Bob S-C...	4/81/4-5
Interpreter for Using Applesoft ROMs from Assembly Language..	Bob S-C...	11/81/2-13
Line Editing Aid.....	Sandy Mossberg...	12/81/11-14
Mini-Assembler for 6502 Written in Applesoft.....	Bob S-C...	7/83/2-7
Patch Applesoft for Garbage Collection Indicator...	Lee Meador...	3/83/22
Relocatable Ampersand Vector.....	Steve Mann...	9/82/15-18
REPEAT and UNTIL for Applesoft.....	Bobby Deen...	11/82/24-28
Save Garbage by Emptying Arrays.....	Bob S-C...	12/82/22-25
Splitting Strings to Fit Your Display.....	Bob S-C...	12/82/26-28
String SWAP Subroutine.....	Bob S-C...	2/81/14-15
Substring Search Function.....	Bob S-C...	4/81/18-20
TRAPPER: An Applesoft Input Tuner.....	Allen Marsalis...	2/83/18-23
Using Applesoft ROMs from Assembly Language		
(FP Arithmetic and Formatted Print).....	Bob S-C...	11/81/2-13
Using USR for a WEEK (2-Byte PEEK).....	Bob S-C...	10/82/30
Variable Cross Reference Program.....	Bob S-C...	11/80/2-8
Arithmetic		
Adding Decimal Values from ASCII Strings.....	Bob S-C...	1/83/21
Really Adding ASCII Strings.....	Bob S-C...	2/83/2-11
Converting Binary Values to Decimal for Printing.....	Bob S-C...	6/83/11-13
Division.....	Bob S-C...	3/83/15-21
On Dividing by Ten.....	Jim Church...	2/82/24
Floating Point Number Format.....	Bob S-C...	11/81/2

Arithmetic, contd.

How To Add and Subtract One.....	Bob S-C...	10/80/2
Multiplying on the 6502.....	Bob S-C...	2/81/11-12
More About Multiplying on the 6502.....	Bob S-C...	6/81/5-8
Using Applesoft ROMs for Arithmetic.....	Bob S-C...	11/81/2-13
Assembler Directives, A Directory of.....	Bob S-C...	9/82/3-14

BBBB

Beginner's Tutorials

Adding Decimal Values from ASCII Strings.....	Bob S-C...	1/83/21
Really Adding ASCII Strings.....	Bob S-C...	2/83/2-11
Base Address Calculations.....	Bob S-C...	9/83/18-21
Bubble Sort Demonstration Program.....	Bob S-C...	6/82/11
Chart of 6502 Operations.....	Bob S-C...	5/81/10
Converting Binary Values to Decimal for Printing.....	Bob S-C...	6/83/11-13
Division.....	Bob S-C...	3/83/15-21
Don't Be Shiftless.....	Bob S-C...	5/81/6-9
How to Add and Subtract One.....	Bob S-C...	10/80/2
How to Move Memory.....	Bob S-C...	1/81/2-6
Jump Vectoring.....	Bob S-C...	9/83/2-8
Loops.....	Bob S-C...	11/81/19-20
Multiplying on the 6502.....	Bob S-C...	2/81/11-12
More About Multiplying on the 6502.....	Bob S-C...	6/81/5-8
On Dividing by Ten.....	Jim Church...	2/82/24
Programming a Language Card.....	Bill Morgan...	1/83/25-26
Search and Perform Subroutine.....	Bob S-C...	8/82/2-5
Simple Hi-Res Animation.....	Mike Laumer...	7/82/15-22
Text File I/O in Assembly Language Programs.....	Bob S-C...	4/81/2-4
Yes/No Subroutine.....	Bob S-C...	6/82/13

Book Reviews

Apple Graphics & Arcade Game Design, Jeffrey Stanton..	Bob S-C...	8/82/23
Apple II Circuit Description, Winston Gayler.....	Bill Morgan...	4/83/20-22
Apple Machine Language, Inman.....	Bob S-C...	3/81/1
Apple Machine Language, Inman.....	Bob S-C...	8/81/6-7
Assembly Lines: The Book, Roger Wagner.....	Bob S-C...	5/82/1
Bag of Tricks, Worth & Lechner.....	Bob S-C...	4/82/1
Beneath Apple DOS, Worth & Lechner.....	Bob S-C...	6/81/19-20
Enhancing Your Apple II, Don Lancaster.....	Bill Morgan...	12/82/29-30
Hardcore Magazine.....	Bob S-C...	9/82/19
Still More on Hardcore Magazine.....	Bob S-C...	1/83/23
INDEX, Bill Wallace.....	Bob S-C...	1/82/12
Micro Cookbook Volume I, Don Lancaster.....	Bill Morgan...	1/83/8
Practical Microcomputer Programming: the 6502, W.J. Weller.....		3/81/1
The Book of Apple Software 1983.....		1/83/26
The Other Epson Manual, Bill Parker.....	Bob S-C...	3/82/15
More about "Other Epson Manual".....	Bill Parker, Bob S-C...	7/82/3
What's Where in the Apple, 2nd Edition.....		1/83/23
6502 Assembly Language Subroutines, Lance Leventhal.....		3/82/23

CCCC

Clarification on Loading the RAM Card.....	Paul Schlyter...	12/82/32
Clear Text Screen Three Times Faster.....	Bob S-C...	9/82/25-27
Converting ToolKit Source to S-C.....	Bob S-C...	10/82/21-27

Cross Assemblers

1802 Version Ready..... 4/83/1
Special Note for 6800 Version.....Bob S-C... 9/82/30
Funny Opcode Names in the 6801 Manual.....Bob S-C... 1/83/9

DDDD

Date Processing Modules.....Brooke Boering... 4/83/13-19

Directives

Allow List of Expressions with .DA Directive.....Bob S-C... 12/80/9
Directives Used in Roger Wagner's Book.....Bob S-C... 9/82/14-15
Directory of Assembler Directives.....Bob S-C... 9/82/3-14
Filler Byte for .BS Directive.....Bill Morgan... 8/83/22
Large Source Files with .IN and .TF Directives....Bill Morgan... 8/82/25-27
Making Lower Case Work in .AS and .AT Strings.....Bob S-C... 8/82/28
Patch to Extend .TF to 63 Target Files.....Bob S-C... 2/83/17
Patch to Fix .TI Problem.....Mike Laumer... 2/83/15
Problem with .IN Directive.....Bob S-C... 11/80/1
.US Directive as Fancier .AS Directive.....Bob S-C... 9/81/12-15

Disassemblers

Broderick's Disassembler (first Ad)..... 8/81/8
Decision System's Disassembler (first Ad)..... 12/80/10
Lee Meador's Disassembler..... 2/81/16
Poor Man's Disassembler...James O. Church..... 11/81/14-17
Rak-Ware's Disassembler (first Ad)..... 12/80/15

DOS 3.2.1 Commented Listings

\$B800-\$BCFF (Disk I/O).....Bob S-C... 5/81/12-20
\$BD00-\$BE9F (RWTS).....Bob S-C... 3/81/15-19
\$BEA0-\$BFFF (Format).....Bob S-C... 4/81/11-14

DOS 3.3 Commented Listings

Boot ROM on Controller Card.....Bob S-C... 8/81/17-20
\$B052-\$B0B5 and \$B35F-\$B7FF (part of File Manager)...Bob S-C... 10/81/18-24
\$B800-\$BCFF (Disk I/O).....Bob S-C... 6/81/10-18
\$BD00-\$BEAE (RWTS).....Bob S-C... 9/81/16-20
\$BEAF-\$BFFF (Format).....Bob S-C... 4/81/14-17

DOS Enhancements

Catalog Arranger.....Bill Morgan... 10/82/2-16
An Addition to the CATALOG ARRANGER.....Dave Barkovitch... 1/83/10
And Another Change.....Bill Collins... 1/83/10
A Filename Editor for the CATALOG ARRANGER.....Bill Morgan... 1/83/11-20
On CATALOG ARRANGER and RAM Card DOS.....Chuck Welman... 2/83/14
Dating Files with Applied Engineering TIME II Card....Bob S-C... 3/82/19-22
DOS Error Trapping from Machine Language.....Lee Meador... 2/82/2-10
EXEC without END from Applesoft.....Bob S-C... 11/82/17
Fast LOAD/BLOAD Patches for DOS 3.3...Bob S-C & Paul Schlyter... 4/83/2-8
Firmware Card in Slot 4.....Michael Sanders... 7/81/1
Free Space Patch Compatible with S-C Macro.....Mike Sanders... 8/82/9-10
Handy EXEC Files.....Bob S-C... 1/82/20-21
Hiding Things Under DOS.....Rick Hatcher... 4/81/10
Correction to "Hiding Things...".....Bob S-C... 6/81/5
More about the Firmware Card in Slot 4.....Bob S-C... 9/81/1
New CATALOG Interrupt.....Col. Paul L. Shetler... 9/83/26-27
New Revision of DOS 3.3 -- Patchers Beware.....Bob S-C... 4/83/23
Detail of Differences in New Version.....Bob S-C... 7/83/26-28
Yet Another New Version of DOS 3.3.....Bob S-C... 9/83/16
Quick Way to Write DOS on a Disk.....Bob Perkins... 8/82/24

DOS Enhancements, contd.

Replacing INIT Can Be Dangerous.....Bill Morgan... 6/83/16-17
Restoring Clobbered Page 3 Pointers.....Preston Black... 7/81/9
ROGRAM TOO LARGE???.....Lee Meador... 5/82/28
RWTS Caller.....Bill Morgan... 5/82/20-25
Secret RWTS Caller inside DOS.....Bill Parker... 5/82/2
Speeding Up Text File I/O.....Paul Schlyter... 7/83/10-12
Making Paul's Patches Fit in DOS.....Bob S-C... 7/83/13-17
Text File Display Command for DOS.....Bob S-C... 7/82/23-27
80-Column SHOW Command.....Robert Bragner... 7/83/24

DOS Problems

Explanation of New DOS 3.3 Append Bug.....Tom Weishaar... 7/83/25
Serious Problem in DOS (with IRQ interrupts).....Bob S-C... 1/82/13

EEEE

Enhancements and Patches to S-C Assembler II Version 4.0

Amperсанд Interface for S-C Assembler II.....Bob S-C... 3/81/20
Assembly Source on Text Files.....Bob S-C... 11/80/9-14
Correction to "Assembly Source on Text Files"..... 8/81/16
A Use for the USR Command.....Bob S-C... 11/80/15
Allow List of Expressions with .DA Directive.....Bob S-C... 12/80/9
Block MOVE and COPY for Version 4.0.....Bob S-C... 12/80/11-14
Bug Corrections.....Bob S-C... 2/81/1, 12
EDITASM & COPY on the Language Card.....Chuck Welman... 3/81/12-14
Installing COPY in the Assembler.....Lee Meador... 1/81/9
Controlling Software Configuration.....Don Taylor... 4/82/24-26
EDIT Command for S-C Assembler II.....Mike Laumer... 1/81/10-16
EDITASM & COPY on the Language Card.....Chuck Welman... 3/81/12-14
Leaving the S-C Assembler II.....Bob S-C... 9/81/11
Another Way Out.....James Church... 10/81/1
Problem with .IN Directive.....Bob S-C... 11/80/1
Putting Version 4.0 on the Language Card.....Paul Schlyter... 1/82/15-19
See All Error Messages in One Pass.....Bob S-C... 4/81/6
Stuffing Code in Protected Places.....Bob S-C... 2/81/9
TAB Locations.....Bob S-C... 1/81/1
Typing LOAD with no filename loads cassette.....Bob S-C... 11/80/1
.US Directive as a Fancier .AS Directive.....Bob S-C... 9/81/12-15
Using Lower Case.....Bob Matzinger... 10/80/4,9-10

Enhancements and Patches to S-C Macro Assembler

Another Lower Case Patch for S-C Macro.....Bob S-C... 10/82/32
Assembly Listing into a Text File.....Bill Morgan... 7/83/8-9
Assembly Listings on Text Files.....Bob S-C... 12/82/13
AUTO/MANUAL Toggle Using Ctrl-A.....R. F. O'Brien... 8/82/6-7
Automatic CATALOG via Esc-C.....Bill Morgan... 6/82/23-24
Automatic CATALOG in the Language Card.....Bill Morgan... 10/82/31
Auto-SAVE.....Greg H. Anders... 4/2-9
Bringing Some Patches Together.....Jim Wetzel... 8/83/24-28
Converting ToolKit Source to S-C.....Bob S-C... 10/82/21-27
Easy Automatic SAVE.....Bob and Bill... 6/82/12
EPRQM Version Available for \$64.....Bob S-C... 5/82/1
Free Space Patch Compatible with S-C Macro.....Mike Sanders... 8/82/9-10
Large Source Files with .IN & .TF Directives.....Bill Morgan... 8/82/25-27
Macro/Videx Connection.....Don Taylor... 8/82/11-22
Right Arrow for the Videx Patches.....Mike Laumer... 9/82/29-31
More on the Macro/Videx Connection.....Bill Linn... 2/83/12-13

Enhancements to S-C Macro, contd.

Making Lower Case Work in .AS and .AT Strings.....	Bob S-C...	8/82/28
More Opcodes for S-C Macro Assembler.....	R. F. O'Brien...	7/83/31-32
Moving the Symbol Table.....	Bill Morgan...	11/82/16
Optional Patch for the TEXT/ Command.....	Bob S-C...	3/83/14
Patch to Fix .TI Problem.....	Mike Laumer...	2/83/15
Patch to Extend .TF to 63 Target Files.....	Bob S-C...	2/83/17
Patch to List Object Code Only Inside Macros.....	Bob S-C...	2/83/27
PAUSE Directive.....	Mike Laumer...	5/83/17-19
Right Arrow for the Videx Patches.....	Mike Laumer...	9/82/29-31
Some Patches for the S-C Macro Assembler.....	Bob S-C...	5/82/3-5
Some Small Patches (^A vs ^I, .BS Filler Byte).....	Bill Morgan...	8/83/22
Star-tiling Stunts.....	Bill Morgan & Mike Laumer...	2/83/25-26
Stopping After One Error.....	Bob S-C...	8/82/27
Toggling Upper/Lower Case in S-C Macro.....	Steven Mann...	12/82/19-20
EPROM Blaster Defined.....	Bob S-C...	3/82/9

FFFF

FLASH!, An Integer BASIC Compiler, Review.....	Bobby Deen...	1/82/22-24
Formatting for Printing		
Converting Binary Values to Decimal for Printing.....	Bob S-C...	6/83/11-13
Dashed Line Across Screen.....	Horst Schneider...	1/83/20
Date Processing Modules.....	Brooke Boering...	4/83/13-19
Formatted Print Routine for Applesoft.....	Bob S-C...	11/81/6-13
General Message Printing Subroutine.....	Bob S-C...	10/80/2-8
Print 2 Bytes in Hex.....	Bob S-C...	12/82/30
Splitting Strings to Fit Your Display.....	Bob S-C...	12/82/26-28
FID -- A Wild-carded Catalog.....	Lee Meador...	8/81/10

GGGG

Generate Machine Code with Applesoft.....	Bob S-C...	9/83/10-12
Graphics		
A New Hi-res Function (HXPLLOT).....	Mike Laumer...	6/82/7-10
Base Address Calculations.....	Bob S-C...	9/83/18-21
Displaying Character Generator EPROMs.....	Bob S-C...	5/83/2-10
Hi-Res SCRNM Function.....	Bob S-C...	5/81/2-3
Hi-Res SCRNM Function with Color.....	David Doudna...	1/82/2-4
Hi-Res Subroutines.....	Bob S-C...	12/81/2-4
Number Nine Graphics Card (description).....	Richard Fabbri...	1/83/28
Simple Hi-Res Animation.....	Mike Laumer...	7/82/15-22
"Apple Graphics & Arcade Game Design", Book Review.....	Bob S-C...	8/82/23

HHHH

Hardware Reviews

Apple //e Notes.....	Bob S-C...	2/83/16-17
More On //e.....	Bob S-C...	3/83/26
Ashby's Easy Shift-Key Modifier.....	Bob S-C...	4/82/13
Axlon RAMDISK 320.....	Bill Morgan...	7/82/11-13
EXCEL-9: 6809 Card with FLEX.....	Bob S-C...	12/81/1
Number Nine Graphics and Processor Cards.....	Richard Fabbri...	1/83/28
PRAWM Board from Advanced Peripheral Enterprises.....	Bob S-C...	4/83/28
Promette from Computer Micro Works.....	Bob S-C...	1/83/28
Some New Cards.....	5/83/20	
Some More 68000 Boards.....	8/83/23	
Tiniest Motherboard, Douglas Electronics.....	Bob S-C...	6/83/15

Hardware Reviews, contd.

Track Balls, Wico and TG Products.....Bill Morgan... 6/83/24-28
Volume Control for Apple Speaker..... 8/83/28
Zero-Insertion-Force Game Socket Extender.....Bob S-C... 2/83/1

IIII

Input Routines

Binary Keyboard Input.....Bob S-C... 8/81/3-4
Blinking Underline Cursor.....Bill Linn... 8/82/29-31
A Note on "Blinking Underline Cursor".....Bob S-C... 9/82/32
Date Processing Modules.....Brooke Boering... 4/83/13-19
Fast String Input Routine.....Bob S-C... 4/81/6-8
Correction to "Fast String Input".....Bob S-C... 10/81/18
Improved "Fast String Input".....Bob S-C... 12/81/16-17
Field Input Routine.....Bob Potts... 9/81/2-7
Integer Input (0-65535) Using Applesoft ROMs.....Peter Meyer... 12/81/insert
Lower-Case Input Using the Shift-Key Mod.....Bob S-C... 6/82/16-17
Numeric Keypad, Simulated.....Bob S-C... 11/80/15-16
TRAPPER: An Applesoft Input Tuner.....Allen Marsalis... 2/83/18-23
Yes/No Subroutine.....Bob S-C... 6/82/13

Integer BASIC

FLASH! Compiler for Integer BASIC, A Review.....Bobby Deen... 1/82/22-24
Pretty Lister for Integer BASIC Programs.....Bob S-C... 12/80/3-8

LLLL

Language Card

Clarification on Loading the RAM Card.....Paul Schlyter... 12/82/32

Lower Case

Lower-Case Input Using the Shift-Key Mod.....Bob S-C... 6/82/16-17
Lower Case Apple.....Bob Matzinger... 7/81/2-5
Patches to S-C Macro Assembler Version 1.1
To Accept ".em" and ".eM".....Bob S-C... 10/82/32
To Generate Lower Case in .AS and .AT Strings.....Bob S-C... 8/82/28
Toggling Upper/Lower Case in S-C Macro.....Steven Mann... 12/82/19-20
Using Lower Case.....Bob Matzinger... 10/80/4,9-10

MMMM

Macros

An "ORG" Macro for Self-Aligning Code.....Bob S-C... 4/83/10-12
Funny Opcode Names in the 6801 Manual.....Bob S-C... 1/83/9
Giant Macro for Writing Messages.....Robert B. Steiner... 7/82/6-7
Macro Branch Library.....R. F. O'Brien... 5/82/29-31
Macros Can Build Macros.....Mike Laumer... 3/83/10
Macro to Speed Up Prime Benchmark.....Anthony Brightwell... 11/82/11-15
Macros for BLT and BGE.....Bob S-C... 1/83/9
Patch to List Object Code Only Inside Macros.....Bob S-C... 2/83/27
Recursive Macro for Repeating Code.....Lee Meador... 4/82/22
Recursive Macro for Generating Data.....Lee Meador... 5/82/17-18
So You Never Need Macros!.....Bob S-C... 10/82/17-18
Using Macros and Nested Macros.....Art Schumer... 4/82/17
Mini-Assembler for 6502 Written in Applesoft.....Bob S-C... 6/83/2-7

Monitor Enhancements

Add Bit Control to Apple Monitor.....Bob S-C... 12/82/10-11
Control-Y Linkage Explained.....Bob S-C... 10/81/14-17
Disassembly of an Address Range.....Bob S-C... 10/81/14-17

Monitor Enhancements, contd.

Dump in both Hex and ASCII

A Beautiful Dump.....	Robert H. Bernard...	3/81/2-5
Adding ASCII to Monitor Dump.....	Bob S-C...	12/81/18-21
Examiner.....	Bill Morgan...	6/82/25-27
Extending the Apple Monitor.....	Bob S-C...	16/81/14-17
Revised Monitor Patch for ASCII Display.....	Bob S-C...	7/83/20-21
80-Column ASCII Monitor Dump.....	Mike Dobe...	9/83/27-28
FADD -- Find Address References.....	Brooke Boering...	5/83/21-23
Lower Case Apple.....	Bob Matzinger...	7/81/2-5
Some Seed Thoughts on Extensions.....	Sanford Greenfarb...	1/83/27

NNNN

New Products

Applesoft Source, On Disk, Completely Commented.....	Bob S-C...	12/82/15
Apple /// Version of S-C Macro Assembler Coming.....		1/83/1
Apple /// Version Working and Ready.....		2/83/1
BASIS 108 Version of S-C Macro Assembler Now Available.....		1/83/1
Bobby Deen's Latest Stuff (Music and Othello).....	Bob S-C...	7/83/32

Cross Assemblers

6809 Version 4.0.....	Chris Wiggs...	10/81/12
Macro 1802 Ready.....		4/83/1
Macro 6800, 6809, and Z-80 Versions.....		8/82/1
Macro 68000 Version.....		9/82/2
Macro PDP-11 Available.....		5/83/1
ES-CAPE, A New Software Tool.....	Bob S-C...	7/82/1
New Compiler: FLASH!.....	reviewed by Bobby Deen...	1/82/22-24
Note about FLASH!.....		3/83/8
S-C Assembler II Version 4.0.....	Bob S-C...	10/80/4-8
S-C Cross Reference Utility.....	Mike Laumer...	4/83/12
S-C Macro Assembler Version 1.0.....	Bob S-C...	3/82/3-7
S-C Macro in EPROM for \$64.....	Bob S-C...	5/82/1
S-C Macro Assembler Version 1.1.....		3/83/1, 24-25
Screen-Oriented Editor for S-C Macro Assembler....	Mike Laumer...	3/83/1
S-C Word Processor.....	Bob S-C...	2/83/28
Note on S-C Word Processor.....	Mike Laumer...	5/83/10
Capture, A Modem Program for S-C Word.....	Jim Church...	5/83/13-15
Source Code for a Word Processor.....	Bob S-C...	2/83/28
Source Code on Disk for Version 4.0.....	Bob S-C...	10/81/1
SynAssembler.....		9/82/2
Vinyl Diskette Pages for S-C Assembler Binder.....	Bob S-C...	5/82/5

Noises and Other Sounds

Alarm in only Eleven Bytes.....	Bob S-C...	2/83/14
A Sight of Sound.....	Herbert A. & Herbert L. McKinstry...	11/82/2
Funny Noise.....	Bob S-C...	4/82/27
My Own Little Bell.....	Bob S-C...	6/82/14
Simple tone, bell, machine-gun, laser-swoop, laser-blast, inch-worm, touch-tones, and Morse code.)...		
.....	Bob S-C...	2/81/2-9
Speaking of Speech.....	Bill Morgan...	11/82/9
Two Fancy Tone Generators.....	Mark Kriegsman...	6/81/2-4
Your Apple Can Talk.....	Bob S-C...	11/82/2-9
Numeric Key Pad, Simulated.....	Bob S-C...	11/80/15-16

PPPP

Paddles and Buttons

Conquering Paddle Jitter.....Brooke Boering... 5/81/4-5
No More Paddle Interaction.....Mike Laumer... 9/82/21-23
Reading the Game Buttons.....Jim Kassel... 5/82/26-28
Reading Two Paddles at the Same Time.....Bob S-C... 3/82/1,24
PATCHER: A General-Purpose Patch Installer.....Bill Morgan... 4/83/24-27

Patches and Modifications to Other Software

Add a New Feature to ES-CAPE.....Bill Linn...12/82/14
Rak-Ware's DISASM and the //e.....Bill Morgan... 4/83/1
Patches for Applewriter to Unhook PLE.....Bob S-C... 2/82/21-22
Using QUICKTRACE with S-C Assembler.....Bob Urschel... 9/83/8

Prime Number Sieve Benchmark

Sifting Primes Faster and Faster.....Bob S-C...10/81/2-10
Even Faster Primes.....Charles Putney... 2/82/15-17
Even Faster Primes Than Charlie's.....Anthony Brightwell...11/82/11-15
Short Note About Prime Benchmarks.....Frank Hirai... 3/83/21

Printer Handler with FIFO Buffer.....Jim Kassel... 2/82/18
Correcton to "FIFO".....Bill Morgan... 3/82/9

Printer Interfaces

80 Columns with Apple's Parallel Interface.....Bob S-C... 4/81/1
Another Way to Get 80 Columns.....Bob S-C... 6/81/1
Improving the Epson Controller Card.....Peter C. Bartlett, Jr... 2/82/11-13
More about the Epson Interface.....Jim Church... 3/82/14
Potential Trouble in TYMAC Interface.....Robert H. Bernard... 4/82/15
The Other Epson Manual -- A Review.....Bob S-C... 3/82/15
More about "Other Epson Manual".....Bill Parker, Bob S-C... 7/82/3

RRRR

Random Number Generator.....Bob S-C... 8/81/11-14
Reviews, see "Book Reviews", "Hardware Reviews", "Software Reviews"

SSSS

S-C Software Corporation

Answered Prayer.....Bob S-C... 7/83/23-24
At Apple/Fest in Houston.....Bob S-C...11/82/1
Burglary, Breaking and Entering.....Bill Morgan... 6/83/9
Trip to California.....Bob S-C...10/82/1
Who Are We and What Are We Doing?.....Mike Laumer... 7/82/4

Screen Printer.....Bob S-C... 7/81/5-7
Epson MX-80 Text Screen Dump.....Ulf Schlichtman... 3/83/12-14
Generic Screen Dump.....Steve Knouse... 9/83/22-24
Scrolling Faster.....Bob S-C... 9/82/25-27
Correction to Bob's Fast Screen Scroll.....Jim Church...10/82/28
Super Scroller.....Jeffrey Scott... 1/83/2-7

Software Reviews

AED II -- A New Applesoft Program Editor.....Bob S-C... 4/82/10-11
New AED Features.....Bob S-C... 5/82/15
Amper-Magic.....Bob S-C... 3/82/10-13
Bag of Tricks.....Bob S-C... 4/82/1
DFX: DOS File Exchange via Hayes Micromodem.....Bill Morgan... 6/82/12
FLASH!, An Integer BASIC Compiler.....Bobby Deen... 1/82/22-24
Hierographic Transport.....Mike Laumer... 7/82/28-31
Intelligent Disassemblers.....Bob S-C...12/80/2
Lee Meador's Disassembler (ad).....Lee Meador... 2/81/16

Software Reviews, contd.

QuickTrace.....	Bob S-C...	8/82
S-C Macro Assembler Version 1.0.....	Bob S-C...	3/82/3-7
S-C Macro Assembler Version 1.1.....	Bill Morgan...	3/83/24-25
The Visible Computer: 6502, Software Masters.....	Bob S-C...	3-83/27-28
Search for Page-Zero References.....	Bob S-C...	6/82/19-21
Sorting Out Page Zero References.....	Tracy L. Shafer...	7/82/10

Source Code On Disk

For All Code Printed in Apple Assembly Line.....	12/80/1
For Applesoft ROM Image.....	12/82/15
For Chris Wiggs' 6809 Cross Assembler (Version 4.0).....	10/81/12
For Laumer Research Screen Editor.....	3/83/1
For S-C Assembler II Version 4.0.....	10/81/1
For S-C Cross Reference.....	4/83/12
For S-C Word Processor.....	2/83/28
Spiral Screen Clear.....	Bob S-C & Roger Keating... 6/83/2-8
Speeding Up Spirals.....	Bob S-C & Charles Putney... 8/83/13-15
Generate Machine Code with Applesoft.....	Bob S-C... 9/83/10-12
Step-Trace Utility.....	Bob S-C... 7/81/11-20

Strings

Adding Decimal ASCII Strings.....	Bob S-C... 2/83/2-11
Fast String Input Routine.....	Bob S-C... 4/81/6-8
Correction to "Fast String Input".....	Bob S-C... 10/81/18
Improved "Fast String Input".....	Bob S-C... 12/81/16-17
Splitting Strings to Fit Your Display.....	Bob S-C... 12/82/26-28
String SWAP Subroutine.....	Bob S-C... 2/81/18-20
Substring Search Function.....	Bob S-C... 4/81/18-20

TTTT

Techniques

All About PTRGET and GTARYPT.....	Bob S-C... 3/83/2-9
Benchmarking Block Moves.....	William R. Savoie... 5/82/7-14
Blinking Underline Cursor.....	Bill Linn... 8/82/29-31
A Note on "Blinking Underline Cursor".....	Bob S-C... 9/82/32
Controlling Software Configuration.....	Don Taylor... 4/82/24-26
Displaying Character Generator EPROMs.....	Bob S-C... 5/83/2-10
Examiner.....	Bill Morgan... 6/82/25-27
Generating Parity.....	Bob S-C... 5/83/24-26
Handling Jump Tables on the Stack.....	Bob S-C... 10/80/11
Implementing New Opcodes Using "BRK".....	Bob S-C... 6/82/2-5
Lower-Case Input Routine Using the Shift-Key Mod.....	Bob S-C... 6/82/16-17
Making Internal JMPs and JSRs Relocatable.....	Peter Meyer... 12/82/2-8
No More Paddle Interaction.....	Mike Laumer... 9/82/21-23
PATCHER: A General-Purpose Patch Installer.....	Bill Morgan... 4/83/24-27
Programming a Language Card.....	Bill Morgan... 1/83/25-26
Random Number Generator.....	Bob S-C... 8/81/11-14
Reading the Game Buttons.....	Jim Kassel... 5/82/26-28
Reformatting a Lot of Text.....	Bob S-C... 6/83/19-22
Run-Anywhere Subroutine Calls.....	Bob Nacon, Bob S-C... 7/82/2
Correction to "Run-Anywhere" re BIT instruction.....	Bob S-C... 8/82/24
Reversing, Getting, and Putting Nybbles.....	Bob S-C... 8/83/19-21
RWTS Caller.....	Bill Morgan... 5/82/20-25
Search and Perform Subroutine.....	Bob S-C... 8/82/2-5
Simple Hi-Res Animation.....	Mike Laumer... 7/82/15-22

Techniques, contd.

Some Fast Screen Tricks.....	Bob S-C...	9/82/25-27
Correction to Bob's Fast Screen Scroll.....	Jim Church...	10/82/28
So You Never Need Macros!.....	Bob S-C...	10/82/17-18
TIME II Card, Applied Engineering, Using the.....	Bob S-C...	3/82/19-22
Tricky Code that Always Skips.....	Bob S-C...	3/82/17-18
Two Ways to Compare a Byte.....	Lee Meador...	8/81/9
Using the Shift-Key Mod.....	Bob S-C...	6/82/16-17
Using Auxiliary Memory in the //e.....	David C. Johnson...	8/83/2-12
What Does This Code Do?.....	John Broderick...	8/81/15
Yes/No Subroutine.....	Bob S-C...	6/82/13
Tips and Hints		
Alarm in only Eleven Bytes.....	Bob S-C...	2/83/14
Dashed Line Across Screen.....	Horst Schneider...	1/83/20
Easy Automatic SAVE in S-C Macro Assembler.....	Bob and Bill...	6/82/12
Merge Selected Bits.....	Bob S-C...	12/82/13
Print 2 bytes in hex.....	Bob S-C...	12/82/30
Saving Source with Apple's Mini-Assembler.....	Jim Church...	9/83/21
Shift 2-byte value right with sign extension.....	Bob S-C...	12/82/15
Star-ting Stunts.....	Mike Laumer & Bill Morgan...	2/83/25-26
Test a Byte without Register Use.....	Bob S-C...	12/82/14

UUUU

Using USR for a WEEK (2-Byte Peek).....	Bob S-C...	10/82/30
---	------------	----------

Utility Programs

Applesoft Program Locator.....	Bill Morgan...	11/82/19-22
Catalog Arranger Utility.....	Bill Morgan...	10/82/2-16
An Addition to the CATALOG ARRANGER.....	Dave Barkovitch...	1/83/10
And Another Change.....	Bill Collins...	1/83/10
A Filename Editor for the CATALOG ARRANGER.....	Bill Morgan...	1/83/11-20
On CATALOG ARRANGER and RAM Card DOS.....	Chuck Welman...	2/83/14
Displaying Character Generator EPROMs.....	Bob S-C...	5/83/2-10
Examiner.....	Bill Morgan...	6/82/25-27
FADD -- Find Address References.....	Brooke Boering...	5/83/21-23
Mini-Assembler for 6502 Written in Applesoft.....	Bob S-C...	6/83/2-7
PATCHER: A General Purpose Patch Installer.....	Bill Morgan...	4/83/24-27
Pretty Lister for Integer BASIC.....	Bob S-C...	12/80/3-8
Relocator.....	Bob S-C...	1/82/8-12
Screen Printer.....	Bob S-C...	7/81/5-7
Epson MX-80 Text Screen Dump.....	Ulf Schlichtman...	3/83/12-14
Generic Screen Dump.....	Steve Knouse...	9/83/22-24
Search for Page-Zero References.....	Bob S-C...	6/82/19-21
Sorting Out Page-Zero References.....	Tracy L. Shafer...	7/82/10
Step-Trace Utility.....	Bob S-C...	7/81/11-20
Corrections to "Step-Trace Utility".....	Bob S-C...	1/82/6
Variable Cross Reference for Applesoft.....	Bob S-C...	11/80/2-8

VVVV

Variable Cross Reference for Applesoft.....	Bob S-C...	11/80/2-8
Corrections to VCR.....	Bob S-C...	7/81/10
Tinkering with Variable Cross Reference.....	Louis Pitz...	9/83/17
Videx/Macro Connection.....	Don Taylor...	8/82/11-22
Right Arrow for the Videx Patches.....	Mike Laumer...	9/82/29-31
More on the Macro/Videx Connection.....	Bill Linn...	2/83/12-13

WWW

WICO Track Ball Driver and Demonstration.....Bill Morgan... 6/83/24-28
WICO Price Increase.....Bill Morgan... 7/83/21
Writing for Apple Assembly Line.....Bob S-C...10/82/32

ZZZZ

Zero-Page References

Searching for All Page-Zero References.....Bob S-C... 6/82/19-21
Sorting Out Page-Zero References.....Tracy L. Shafer... 7/82/10

6502

Chart of 6502 Operations.....Bob S-C... 5/81/10
Hardware Error in ALL 6502 Chips!.....Bob S-C...10/80/10,11
Mini-Assembler for 6502 Written in Applesoft.....Bob S-C... 7/83/2-7
So-called Unused Opcodes.....Bob S-C... 3/81/7-11
6500/1 One-Chip Computer.....Dan Pote... 1/82/21

65C02

New Enhanced 6502 Nearly Here!.....Bob S-C...12/82/16-17
Note on GTE Version.....Don Lancaster... 6/83/1
Latest 65C02 Word.....Bill Morgan... 7/83/1,18
65C02: What It Is.....Bill Morgan... 8/83/12
65C02 Notes, BIT Immediate Opcode Faulty.....Bill Morgan... 9/83/1

Apple Assembly Line is published monthly by S-C Software Corporation, P.O. Box 280300, Dallas, Texas 75228. Phone (214) 324-2050. Subscription rate is \$18 per year in the USA, sent Bulk Mail; add \$3 for First Class postage in USA, Canada, and Mexico; add \$12 postage for other countries. Back issues are available for \$1.50 each (other countries add \$1 per back issue for postage).

Every three months the source code is collected into a Quarterly Disk. The quarters are Jan-Mar, Apr-Jun, Jul-Sep, and Oct-Dec. Each Quarterly Disk costs \$15.00.

All material herein is copyrighted by S-C SOFTWARE CORPORATION, all rights reserved.

(Apple is a registered trademark of Apple Computer, Inc.)

Price Changes.....Bob Sander-Cederlof

It has been nearly two years since we raised the price of a subscription from \$12 to \$15 per year, and now we are forced to another increase. Effective January 1, 1984, a year's subscription by bulk mail in the USA will be \$18. For First Class Mail in the USA, Canada, and Mexico, add \$3. Subscriptions to other countries, including postage, will be \$30 per year.

You can beat the price by renewing early. All renewals received before January 1st will be at the old rates.

Now for some good news! We want to reduce our inventory of back issues, so we are offering some special prices. We normally sell them for \$1.50 each; between now and January 1st you can buy them for only \$1 each!

We want to encourage more of you to save your time and energy by getting the Quarterly Disks, with all the source code from three issues already correctly entered. Each Quarterly Disk costs only \$15. To save even more trouble, and some \$\$\$, you can subscribe to the Quarterly Disks. Effectively immediately, prepaid subscriptions for four Quarterly Disks will be only \$45. You save 25%!

Continuing in the Christmas spirit, here are some more specials good through the end of this year, only for subscribers to Apple Assembly Line:

	Regular	Special
FLASH! Integer BASIC Compiler	\$79	\$50
The Visible Computer: 6502	\$50	\$40
ES-CAPE	\$60	\$40
S-C Math Disk & Game Disk Set	\$35	\$20
Laumer's Full Screen Editor	\$49	\$40

New Basis Version 1.1 Available

If any of you are using the S-C Macro Assembler with a Basis 108 computer, Bob Matzinger has adapted version 1.1 for you. Call us for the upgrade price. (214) 324-2050.

Duplicated Ideas and Red Faces.....Bob Sander-Cederlof

I suppose it had to happen at least once in three years, but it still came as a shock.

Last June I wrote and published a program and article called Amper-Monitor, and then I did it all over again for the September issue. The programs are slightly different, both in design and implementation, but they still do the same thing.

Maybe now that we have a complete index to the first three volumes I won't make this mistake again.

Faster Booting for ScreenWriter II.....Bob Leedom
Glenwood, Maryland

I have found a solution to ScreenWriter II's long boot-up time (which is one of my few complaints with the product). Would you believe a reduction from 46 seconds to just under 14 seconds?

The solution was given in a patch to DOS 3.3 given by Paul Schlyter and Bob Sander-Cederlof in the April 1983 issue of AAL. Since ScreenWriter's DOS is nearly identical to 3.3, I was inspired to try the patch (on ONE of my two copy-protected original disks) -- and it worked!

I installed the patch between lines 50 and 60 of APP2 (ScreenWriter's customizeable startup program). The POKes will only be performed at startup -- if you look closely at APP2, you'll see that the POKEing lines will be skipped when the program is used to switch between Editor and Runoff in the non-RAMcard version.

To install the patch, do the following:

1. From BASIC, LOAD APP2
2. Type in Lines 51 - 59, carefully!
3. SAVE APP2
4. RUN CUSTOMIZEA

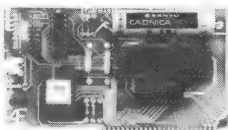
That's it! You will now have a fast-booting ScreenWriter. You may also want to do this to some of your normal DOS 3.3 disks -- the patch is in an unused area of DOS, and seems to coexist happily with everything else I tried (like PLE and GPLE for instance). Exception: in the //e version of DOS 3.3, the patch screws up the infamous APPEND command -- no great loss, in my opinion.

```
51 READ N: IF N=0 THEN 59:REM Make this "THEN 60" (60 is the
    next ScreenWriter II line) when line 59 is DELETED
52 READ A: SUM = SUM + A + N
53 FOR I = 1 TO N: READ P: POKE A,P: A=A+1: SUM=SUM+P: NEXT
54 GOTO 51
55 DATA 44, 47721, 173, 230, 181, 208, 36, 173, 194, 181, 240,
    31, 173, 203, 181, 72, 173, 204, 181, 72, 173, 195, 181,
    141, 203, 181, 173, 196, 181, 141, 204, 181, 32, 182, 176,
    176, 3, 76, 223, 188, 76, 111, 179, 76, 150, 172
56 DATA 33, 48351, 238, 228, 181, 208, 3, 238, 229, 181, 238,
    196, 181, 238, 204, 181, 206, 194, 181, 208, 11, 104, 141,
    204, 181, 104, 141, 203, 181, 76, 150, 172, 76, 135, 186
57 DATA 2, 44198, 105, 186
58 DATA 0
59 IF SUM <> 153114 THEN PRINT "OOPS! DATA IS OFF BY
    "153114-SUM: STOP: REM (Delete this line when you are SURE
    it works!)
```

APPLIED ENGINEERING

THE BEST PERIPHERALS FOR THE BEST COMPUTER

The TIMEMASTER Finally a clock that does it ALL!



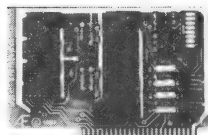
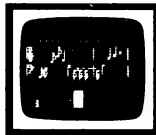
- Designed in 1983 using I.C. technologies that simply did not exist when most other Apple clocks were designed.
- Just plug it in and your programs can read the year, month, date, day, and time to 1 millisecond! The only clock with both year and ms.
- Powerful 2K ROM driver — No clock could be easier to use.
- Full emulation of most other clocks, including Mountain Hardware's Appleclock (but you'll like the TIMEMASTER mode better).
- Basic, Machine Code, CP/M and Pascal software on 2 disks!
- Eight software controlled interrupts so you can execute two programs at the same time. (Many examples are included)
- On board timer lets you time any interval up to 48 days long down to the nearest millisecond.

The TIMEMASTER includes 2 disks with some really fantastic time oriented programs (over 25) plus a DOS dater so it will automatically add the date when disk files are created or modified. This disk is over a \$200.00 value alone — we give the software others sell. All software packages for business, database management and communications are made to read the TIMEMASTER.

If you want the most powerful and the easiest to use clock for your Apple, you want a TIMEMASTER.

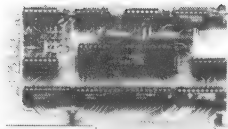
PRICE \$129.00

Super Music Synthesizer



- Complete 16 voice music synthesizer on one card. Just plug it into your Apple, connect the audio cable (supplied) to your stereo, boot the disk supplied and you are ready to input and play songs.
- It's easy to program music with our compose software. You will start right away at inputting your favorite songs. The Hi-Res screen shows what you have entered in standard sheet music format.
- Now with new improved software for the easiest and fastest music input system available anywhere.
- We give you lots of software. In addition to Compose and Play programs, 2 disks are filled with over 30 songs ready to play.
- Easy to program in Basic to generate complex sound effects. Now your games can have explosions, phaser zaps, train whistles, death cries. You name it, this card can do it.
- Four white noise generators which are great for sound effects.
- Plays music in true stereo as well as true discrete quadraphonic.
- Full control of attack, volume, decay, sustain and release.
- Will play songs written for ALF synthesizer (ALF software will not take advantage of all the features of this board. Their software sounds the same in our synthesizer.)
- Automatic shutoff on power-up or if reset is pushed.
- Many many more features.

PRICE \$159.00



Z-80 PLUS!

- TOTALLY** compatible with ALL CP/M software.
- The only Z-80 card with a special 2K "CP/M detector" chip.
- Fully compatible with microsoft disks (no pre-boot required).
- All new 1983 design incorporates the latest in I.C. technologies.

- Red "CP/M WORKING" LED indicator, the Z-80 Plus does not interfere with non-CP/M programs.
 - An on-card PROM eliminates many I.C.'s for a cooler, less power consuming board. (We use the Z-80A at a fast 4MHz)
 - Does EVERYTHING the other Z-80 boards do, plus Z-80 interrupts.
- Don't confuse the Z-80 Plus with crude copies of the microsoft card. The Z-80 Plus employs a much more sophisticated and reliable design. With the Z-80 Plus you can access the largest body of software in existence. Two computers in one and the advantages of both, all at an unbelievably low price.

PRICE \$139.00

COMING SOON: The Z-80 Plus for the Apple III

Viewmaster 80

There used to be about a dozen 80 column cards for the Apple, now there's only ONE.

- TOTALLY** Vindex Compatible
- 80 characters by 24 lines, with a sharp 7x9 dot matrix
- On-board 40/80 soft video switch with manual 40 column override
- Fully compatible with ALL Apple languages and software — there are NO exceptions
- Low power consumption through the use of CMOS devices
- All connections on the card are made with standard video connectors, no cables are soldered to the board
- All new 1983 design (using a new Microprocessor based C.R.T. controller)

JUST COMPARE!

	VIEWMASTER	169	YES	YES	YES	YES	YES	YES	YES	YES
SUPRTERM	175	NO	YES	NO	NO	NO	YES	YES	YES	YES
WIZARD80	245	NO	NO	YES	YES	NO	YES	YES	YES	YES
VISION80	375	YES	YES	YES	YES	NO	NO	NO	NO	NO
OMNIVISION	295	NO	YES	NO	NO	NO	YES	YES	YES	YES
VIEWMAX80	219	YES	YES	YES	YES	NO	NO	NO	YES	YES
SMART80	345	YES	YES	YES	YES	NO	YES	NO	YES	YES
VIDEOTERM	345	NO	NO	NO	YES	YES	YES	NO	YES	YES

The VIEWMASTER 80 works with all 80 column applications including CP/M, Pascal, WordStar, Format II, Easywriter, Apple Writer II, Visicalc, and many others. The VIEWMASTER 80 is THE MOST compatible 80 column card you can buy at ANY price!

PRICE \$169.00

MemoryMaster IIe 128K RAM Card

- Expands your Apple IIe to 128K memory
- Provides an 80 column text display
- Compatible with all Apple IIe 80 column and extended 80 column card software. (Same physical size as Apple's 64K card)
- Available in 64K and 128K configurations
- Bank select LED's for each 64K bank
- Permits your IIe to use the new double high resolution graphics
- Automatically expands Visicalc to 95K storage in 80 columns! The 64K configuration is all that's needed, 128K can take you even higher.

- Complete documentation included, we show you how to use all 128K.
- If you already have Apple's 64K card, just order the MEMORYMASTER with 64K and use the 64K from your old board to give you a full 128K. (The board is fully socketed so you simply plug in more chips.)

MemoryMaster with 128K	\$249
Upgradeable MemoryMaster with 64K	\$169
Non-Upgradeable MemoryMaster with 64K	\$149

Our boards are far superior to most of the consumer electronics made today. All I.C.'s are in high quality sockets with mil-spec. components used throughout. P.C. boards are glass-epoxy with gold contacts. Made in America to be the best in the world. All products work in APPLE IIe, II+, IIx and Franklin (except MemoryMaster). Applied Engineering also manufactures a full line of data acquisition and control products for the Apple. A/D converters and digital I/O cards, etc. Please call for more information. All our products are fully tested with complete documentation and available for immediate delivery. All products are guaranteed with a no hassle **THREE YEAR WARRANTY.**

All Orders Shipped Same Day. Texas Residents Add 5% Sales Tax. Add \$10.00 if Outside U.S.A. Dealer Inquiries Welcome.

Send Check or Money Order to:
APPLIED ENGINEERING
P.O. Box 470301
Dallas, TX 75247

Call (214) 492-2027
7 a.m. to 11 p.m. 7 days a week
MasterCard, Visa & C.O.D. Welcome

Large Assembly Listing into Text File.....Robert F. O'Brien
Dublin, Ireland

I liked the procedure for getting listings into a text file during assembly (AAL July '83). However, it won't work if the file is too large and requires .IN directives. I recently did a large assembly using the following source code:

```
0      .DU
1      .TF LISTING
2      .IN PART1
3      .IN PART2
4      .ED
```

What I expected to get was a 356-sector text file on disk, but all I got was a 2-sector file -- the code for PART1 and PART2 was not sent to the disk (they did list to the screen!)

I first tried to solve my particular problem by making more RAM available for the assembly by moving the Symbol Table base down to \$400. I thought that should work, since I use an 80-column card and not the Apple's text screen. However, the assembler and the system monitor had other ideas, and promptly destroyed the symbol table by scrolling the screen memory.

However, I did manage to get my large assembly listing to go to disk as a text file -- by doing it in two parts. I used a utility program from the assembler disk to give each part the missing label definitions from the other part.

The steps are as follows:

- 1) Assemble the code normally with .IN directives.

```
0001      .IN PART1
0002      .IN PART2
```

- 2) BRUN B.MAKE EQUATE FILE (from the S-C Macro Version 1.1 Disk.) That creates a file of .EQ statements called SYMBOLS which contains all the normal labels and values from the Symbol Table in memory.
- 3) Merge SYMBOLS with PART1 and delete all duplicate labels from the SYMBOLS section.
- 4) Assemble PART1 using the .DU-.TF-.ED trick, and using .LIST OFF/ON so that the SYMBOLS section does not write to the text file.
- 5) Repeat steps 3 and 4 on PART2.

It is a bit laborious deleting all the duplicate labels in the two assemblies. I hope someone can suggest a patch to the assembler to prevent it from reporting "EXTRA DEFINITION ERROR". That certainly would make this listing process easier.

Avoiding EXTRA DEFINITION ERROR.....Bill Morgan

No sooner said...

OK, here are some patches to defeat the check for double definitions in the S-C Macro Assemblers. Just put an RTS (\$60) at the appropriate location:

Version 1.0 -- Motherboard: \$221D
Language Card: \$E369

Version 1.1 -- Motherboard: \$210E
Language Card: \$E228

Be very certain that any double definitions are intentional and identical. If you use the same label with two different values (unless it's defined with .SE) the assembler cannot produce correct code.

Lower Case Titles in Version 1.1.....Bob Matzinger

A simple one-byte patch will enable you to use lower-case letters inside .TI titles. There are eight versions of the assembler on the Version 1.1 release disk, and the byte to be changed is in a different place for each version.

The code for the .TI directive looks the same wherever it is located. Here is a hex dump of the code, with a square around the byte to be changed:

```
A2 00      LDX #0
20 3E x2   JSR $123E or $D23E
C9 2C      CMP #$2C
D0 0D      BNE ...

20 3E x2   JSR $123E or $D23E

B0 08      BCS ...
9D 70 01   STA $170,X
```

The following table shows the address of the byte to be changed:

File Name	x = 1000	x = D000
	-----	-----
S-C.ASM.MACRO.x	\$2CE6	\$EE00
S-C.ASM.MACRO.x.E	\$2CC2	\$EDDC
S-C.ASM.MACRO.x.STB80	\$2DDA	\$EEFD
S-C.ASM.MACRO.x.VIDEX	\$2DB1	\$EED4

Once you find the right byte, which contains \$3E, change it to \$4E. (Remember to change a byte in the RAM card you need to write-enable it first.)

QUICKTRACE

relocatable program traces and displays the actual machine operations, *while* it is running without interfering with those operations. Look at these **FEATURES**:

Single-Step mode displays the last instruction, next instruction, registers, flags, stack contents, and six user-definable memory locations.

Trace mode gives a running display of the Single-Step information and can be made to stop upon encountering any of nine user-definable conditions.

Background mode permits tracing with no display until it is desired. Debugged routines run at near normal speed until one of the stopping conditions is met, which causes the program to return to Single-Step.

QUICKTRACE allows changes to the stack, registers, stopping conditions, addresses to be displayed, and output destinations for all this information. All this can be done in Single-Step mode while running.

Two optional display formats can show a sequence of operations at once. Usually, the information is given in four lines at the bottom of the screen.

QUICKTRACE is completely transparent to the program being traced. It will not interfere with the stack, program, or I/O.

QUICKTRACE is relocatable to any free part of memory. Its output can be sent to any slot or to the screen.

QUICKTRACE is completely compatible with programs using Applesoft and Integer BASICs, graphics, and DOS. (Time dependent DOS operations can be bypassed.) It will display the graphics on the screen while **QUICKTRACE** is alive.

QUICKTRACE is a beautiful way to show the incredibly complex sequence of operations that a computer goes through in executing a program

QUICKTRACE

\$50

Is a trademark of Anthro-Digital, Inc.

Copyright © 1981

Written by John Rogers

See these programs at participating Computerland and other
fine computer stores.

Anthro - Digital Software, Inc.
P.O. Box 1385 Pittsfield, MA 01202

Suppressing Unwanted Object Bytes

Sometimes we want to get an assembly listing that doesn't use up half a page of paper for each .AS or .HS line, listing three object bytes on each line. A number of you have asked for a patch to show the source line without listing each and every one of those hex bytes.

Well, David Roberts, a subscriber in Australia, has come up with a simple way to do just that. He uses macros! David suggests these definitions:

```
.MA AS
.AS -"j1"
.EM
```

```
.MA AT
.AT "j1"
.EM
```

```
.MA HS
.HS "j1"
.EM
```

Now you can code text with >AS."THIS IS MY STRING", and use the .LIST MOFF option to suppress the hex listing. That's really a "why didn't I think of that?" Thanks, David.

Where To?.....Bill Morgan

The word is that the new Mackintosh machine from Apple is going to be 68000-based and affordable. I know that I am going to want one, and I would like to get a leg up on learning the machine, so I'm starting to study 68000. It looks like a lot of fun. With seventeen registers addressing 16 megabytes at 12 megaHertz or thereabouts, we should be able to do just about anything we want. I'll have a review next month of a new 68000 trainer board for your Apple, at about half the price of the existing 68000 boards.

To get to the point, how many of you good folks out there are interested in 68000? How many of you already know a little or a lot about it? Should we start a new newsletter about Mackintosh? Should we devote a few pages of this one to it? Let us hear from you.

And another thing, how about C language? Several of you have mentioned that great August issue of Byte and expressed an interest in learning more about C. I know that I'm going to study up on it. There is a good C compiler available for the Apple, the Aztec C Compiler System from Manx Software. I'll have a review of it in the next month or two, and we may start carrying it for sale. Let me know if you're interested.

Macro-Calculated Spiral Screen Clear.....Bruce V. Love
Hamilton, New Zealand

Here is what I think is a beautiful example of using nested recursive macros with the new .SE directive to calculate the addresses for a Spiral Screen Clear.

The macro SPIRAL calls, in order, LEFT, BOTTOM, RIGHT, and TOP to produce the code to handle each side of the screen. Each of those macros adjusts the appropriate X or Y coordinate and then calls GETADR to calculate the addresses and actually assemble the next instruction pair.

This program won't win any prizes for fast assembly: I timed it at almost 4 minutes. You could speed up the process by rewriting the BOTTOM and TOP macros. They really don't have to call GETADR for all the calculation, they only need to increment or decrement the addresses, but that destroys the symmetry of the original.

I have also produced a faster version of the program. This one uses self-modifying code to avoid shifting the already-cleared bytes on the screen. It's interesting to watch the self-modifying version accelerate as it moves fewer bytes each time through the loop. To produce the faster version, just replace the code from line 1680 on with this new code:

```

1680 POINTER .EQ 0
1690 *-----
1700     LDY #0          no indexing
1710     LDA #END        start pointer at end of code
1720     STA POINTER
1730     LDA /END
1740     STA POINTER+1
1750 .2   JSR LOOP        do one step
1760     LDA #$AD        restore LDA code
1770     STA (POINTER),Y
1780 *-----
1790     LDA POINTER     decrement pointer
1800     SEC              by 6
1810     SBC #6
1820     STA POINTER
1830     BCS .1
1840     DEC POINTER+1
1850 .1   LDA #$60        insert RTS code
1860     STA (POINTER),Y
1870 *-----
1880     LDA POINTER     compare pointer
1890     CMP #LOOP        to beginning of code
1900     BNE .2
1910     LDA POINTER+1
1920     SBC /LOOP
1930     BNE .2          branch if not yet done
1940 *-----
1950 FIXUP LDA #$AD        restore LDA
1960     STA LOOP          at beginning
1970     LDA #$60          and RTS
1980     STA END           at end
1990     JMP $3D0          and reenter DOS
2000 *-----
2010 SAVE .DA #$A0        <space> to fill screen
2020 *-----
2030 LOOP >SPIRAL
2040     LDA SAVE
2050     STA $5B4
2060 END     RTS

```

```

1000      .TF CLEAR
1010      .LIST OFF
1020      *-----
1030      .MA SPIRAL
1040      >LEFT          move left side up
1050 BOTLFT .SE BOTLFT-1      and move corner up
1060      >BOTTOM        move bottom left
1070 BOTRGT .SE BOTRGT-1      and move corner left
1080      >RIGHT          move right side down
1090 TOPRGT .SE TOPRGT+1      and move corner down
1100      >TOP            move top right
1110 TOPLFT .SE TOPLFT+1      and move corner right
1120      .DO TOPLFT<13     done?
1130      >SPIRAL          no, do it again
1140      .FIN
1150      .EM
1160      *-----
1170      .MA GETADR
1180 ADRTO  .SE ADRFRM
1190 BLOCK  .SE Y.CORD/8      hi, mid, or low, 0-2
1200 BLK.AD .SE BLOCK*28      block offset
1210 TEMP   .SE BLOCK*8
1220 LINE   .SE Y.CORD-TEMP   line within block, 0-7
1230 LIN.AD .SE LINE*80        line offset
1240 ADRFRM .SE $400+BLK.AD+LIN.AD+X.CORD
1250      LDA ADRFRM
1260      STA ADRTO
1270      .EM
1280      *-----
1290      .MA LEFT
1300 Y.CORD .SE Y.CORD+1      down one step
1310      >GETADR
1320      .DO Y.CORD<BOTLFT  done?
1330      >LEFT            no, again
1340      .FIN
1350      .EM
1360      *-----
1370      .MA BOTTOM
1380 X.CORD .SE X.CORD+1      right one step
1390      >GETADR
1400      .DO X.CORD<BOTRGT  done?
1410      >BOTTOM          no, again
1420      .FIN
1430      .EM
1440      *-----
1450      .MA RIGHT
1460 Y.CORD .SE Y.CORD-1      up one step
1470      >GETADR
1480      .DO Y.CORD>TOPRGT  done?
1490      >RIGHT            no, again
1500      .FIN
1510      .EM
1520      *-----
1530      .MA TOP
1540 X.CORD .SE X.CORD-1      left one step
1550      >GETADR
1560      .DO X.CORD>TOPLFT  done?
1570      >TOP              no, again
1580      .FIN
1590      .EM
1600      *-----
1610 BOTLFT .SE 23            bottom left Y coord
1620 BOTRGT .SE 39            bottom right X coord
1630 TOPRGT .SE 0             top right Y coord
1640 TOPLFT .SE 1             top left X coord
1650 X.CORD .SE 0             start with upper
1660 Y.CORD .SE 0             left corner
1670 ADRFRM .SE $400
1680      *-----
1690      LDX #960            do the loop 960 times
1700      LDY /960
1710      LDA #$A0            put space in center
1720      STA $5B4
1730 LOOP   >SPIRAL          do one spiral
1740 END     DEX
1750      BNE .1              branch if not done
1760      DEY
1770      BPL .1
1780      JMP $3D0            exit to DOS
1790 .1     JMP LOOP          go spiral again

```

Counting Lines.....Bill Morgan

When Bob and I were first looking at Bruce Love's version of the Spiral Screen Clear, we got to wondering just how many lines actually were being processed by the assembler. With all those nested recursive macros, the total was bound to be in the thousands. Here's a little filter program I threw together to do a count:

```

0000-          1000 COUNT.LO .EQ 0
0001-          1010 COUNT.HI .EQ 1
0036-          1020 OUTHOOK  .EQ $36
03EA-          1030 DOSHOOK  .EQ $3EA
              1040 *-----
              1050          .OR $300
              1060
0300- A9 00    1070          LDA #0
0302- 85 00    1080          STA COUNT.LO      zero the counters
0304- 85 01    1090          STA COUNT.HI
0306- A9 11    1100          LDA #LINE.COUNTER
0308- 85 36    1110          STA OUTHOOK      direct output
030A- A9 03    1120          LDA /LINE.COUNTER to my routine
030C- 85 37    1130          STA OUTHOOK+1
030E- 4C EA 03 1140          JMP DOSHOOK
              1150 *-----
              1160 LINE.COUNTER
0311- C9 8D    1170          CMP #$8D          carriage return?
0313- D0 06    1180          BNE .1           no, exit
0315- E6 00    1190          INC COUNT.LO      yes, count it
0317- D0 02    1200          BNE .1
0319- E6 01    1210          INC COUNT.HI
031B- 60      1220 .1      RTS

```

I assembled that code at \$300, and then used these commands to set the PRT vector:

```
: $C083 C083 D009:4C 0 3 N C080
```

(For the motherboard versions of the S-C Assemblers, you only need to type :\$1009:4C 0 3)

With that in place just load a source file, set .LIST ON, type PRT, and then type ASM. When the assembly is finished, type PR#0 to get the output back to the screen. Now you can type :\$0.1 to look at the counters. You might also want to put a .LIST OFF line at the end of your program, so the count won't include the Symbol Table.

By the way, when the macros are expanded those 80 lines of Bruce's program produce 13,593 lines of code, or enough to fill over 200 pages of printout.

With these tools you can really program!

Presenting a selection of the best programming tools for the entire Apple II family!

S-C Macro Assembler - Combined editor/assembler includes 29 commands and 20 directives; with macros, conditional assembly, global replace, edit, and more. Well-known for ease-of-use and powerful features. Thousands of users in over 30 countries and in every type of industry attest to its speed, dependability, and user-friendliness. Blends power, simplicity, and performance to provide the optimum capabilities to both beginning and professional programmers. With 100-page manual and reference card, \$80. (If you own an older version, call for an upgrade price.)

Cross Assembler Modules - Owners of the S-C Macro Assembler may add the ability to develop programs for other systems. We have modules for most of the popular chips, at very reasonable prices:

6800/01/02	\$32.50
6805	\$32.50
6809	\$32.50
68000	\$50.00
Z-80	\$32.50
PDP-11	\$50.00
8048	\$32.50
8051	\$32.50
8080/85	\$32.50
1802/04/05	\$32.50

All of the cross assemblers retain the full power of the S-C Macro Assembler. You can develop programs for burning into EPROMs, transfer through a data-link, or direct execution by some of the plug-in processor cards now on the market.

Apple Assembly Line - Monthly newsletter for assembly language programmers, beginner or advanced. Tutorial articles; advanced techniques; handy utility programs; commented listings of code in DOS and Apple ROMs; reviews of relevant new books, hardware, software; and more! \$15 per year (add \$3 for first class postage in USA, Canada, Mexico; add \$13 postage for other countries).

ES-CAPE (Extended S-C Applesoft Program Editor)
Full-function, interactive program editor for Applesoft.

For painless programming and accurate entry of programs from magazines: fast, easy changes; split-screen operation; global search and replace; automatic line numbers; keyboard macros; and more! \$60

Flash! - The Integer BASIC Compiler from Laumer Research. Transforms your programs into machine language so they run many times faster. Adds many new functions and features to the language, supporting hi-res graphics and more! Optional assembly source code creation for use with S-C Assembler. Requires Integer BASIC in RAM or ROM to edit source programs (not required for compiled programs). Source code of run-time package also available. Compiler, \$79; run-time source, \$39.

Double Precision Floating Point for Applesoft - For the scientist or engineer who is not satisfied with Applesoft's nine-digit precision. With this 2048-byte machine language extension to Applesoft, you can get 21-digit precision whenever you need it! Supports +, -, *, /, INPUT, and PRINT. Subroutines coded in Applesoft for 21-digit precision math functions (sin, cos, sqrt, log, exp, etc.) are also included. And it's fast! Only \$50.

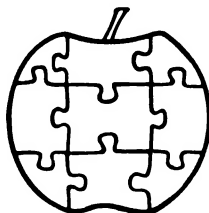
DISASM - Symbolic two-pass disassembler handles data tables, displaced object code, and includes a pre-defined symbol table of your choice. Produces ready-to-assemble source code with labels from machine code in ROM or RAM, formatted for S-C, Lisa, or ToolKit assemblers. \$30.

S-C XREF - Designed to work with the S-C Macro Assembler, produces a cross reference table of all labels used in your assembly source code. Only \$20, or \$50 with complete source code on disk.

All of our products are designed to be used. You can make back-up copies with any standard Apple disk copy program, and copies of the individual files can be moved to your working disks. We trust you.

S-C Software Corporation has frequently been commended for outstanding support: competent telephone help, continuing enhancements, and excellent upgrade policies. Call now and talk to Bob Sander-Cederlof or Bill Morgan. We want to help you program!

S-C Software Corporation
2331 Gus Thomasson, Suite 125
Dallas, Texas 75228
(214) 324-2050



We take Mastercard, Visa, and American Express

Apple is a trademark of Apple Computer

If You Like Shapes, Try Shapemaker.....Bob Sander-Cederlof

Frank Belanger sent me a copy of his new Hi-Res utility program, called SHAPEMAKER. I know, there are a lot of these on the market, such as Accu-Shapes and Apple Mechanic. Frank's is priced between those two, at \$35, and look at all you get:

- * Shape Editor
- * Shape Table Editor
- * Nine &-routines, including
 - * Clear any window in the hi-res page
 - * Display string of shapes
 - * Input anything
- * 44-page manual
- * Source Code of &-routines in S-C format
- * Unprotected, copyable, modifiable

If these features interest you, write Frank at 4200 Avenue B, Austin, TX 78751. Or call (512) 451-6868.

ES-CAPE will set your creativity free!

ES-CAPE will help you develop, enter, and modify Applesoft programs. Even if you are only copying a program from a magazine, ES-CAPE will help you do it three times faster!

Visualize this: by pressing just a key or two, you can...

- See the disk catalog, select a program, and load it into memory.
- Browse through the program a screen or a line at a time.
- Edit lines using powerful commands like the word processors have: insert, delete, truncate, overtype, scan to beginning or end or to a particular character, and more.
- See the values of the variables used by your Applesoft program as it ran.
- Save the modified program. ES-CAPE remembers the file name for you!

ES-CAPE is easy to learn and use!

- Well-written User Manual guides you through the learning process.
- Handy Quick Reference Card reminds you of all features and commands.
- Built-in help screens and menus refresh your memory. You don't have to memorize anything!
- The disk is NOT protected! You can put ES-CAPE on every disk you own, and make as many backup copies as you need.

ES-CAPE will speed up and simplify your Applesoft programming!

- Choose a starting value and step size for automatic line numbering.
- Swiftly find all references to a given variable, line number, or any other sequence of characters.
- Quickly and automatically scan your program for any sequence of characters and replace them with a new spelling.
- Enter commonly used words or phrases with a single keystroke. A full set of pre-defined macros is provided, which you may modify as you wish.
- Display a DOS Command Menu with a single keystroke. A second keystroke selects CATALOG, LOAD, SAVE, and other common DOS commands. You can easily manage a disk-full of programs!

ES-CAPE is available now at many fine computer stores, or directly from S-C Software Corporation. The price is only \$60.

S-C SOFTWARE CORPORATION
2331 Gus Thomasson, Suite 125
Dallas, TX 75228 (214) 324-2050

Professional Apple Software Since 1978

Visa, MasterCard, American Express, COO accepted.

Apple is a trademark of Apple Computer, Inc.



Apple Assembly Line is published monthly by S-C SOFTWARE CORPORATION, P.O. Box 280300, Dallas, Texas 75228. Phone (214) 324-2050. Subscription rate is \$15 per year in the USA, sent Bulk Mail; add \$3 for First Class postage in USA, Canada, and Mexico; add \$13 postage for other countries. Back issues are available for \$1.50 each (other countries add \$1 per back issue for postage).

All material herein is copyrighted by S-C SOFTWARE CORPORATION, all rights reserved. (Apple is a registered trademark of Apple Computer, Inc.)